普通高等教育“十一五”国家级规划教材升级版

计算机英语教程

（第六版）

司爱侠　张强华　主编

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书旨在切实提高读者实际使用英语的能力。本书立足实用，软件、硬件和网络并重，同时兼顾发展热点。

本书体例上以课为单元，课文选材广泛、风格多样；给出课文中出现的新词，读者由此可以积累计算机专业的基本词汇；给出课文中的常用词组；讲解课文中出现的疑难句子；每课一个核心语法点，系统地讲述计算机领域中常见的语法；书中有丰富的练习：既有语法练习，也有针对课文的练习，还有针对“计算机水平考试”的练习；技能训练中模拟了一个工作环境，以训练读者运用语言的能力；阅读材料进一步扩大读者的视野。书后附有“英语基本句型”、“英语单词速记法”和“参考答案”。本书特别配套了朗读资料，可登录电子工业出版社华信教育资源网免费下载，供读者随时播放收听。

本书曾获得全国畅销书（科技类）、江苏省高教科研成果优秀奖，并被教育部批准为“普通高等教育‘十一五’国家级规划教材”。

本书既可作为本科院校、高等专科院校、高等职业院校、中专的计算机专业英语教程，也可供所有IT行业人员自学参考。

图书在版编目（CIP）数据

计算机英语教程/司爱侠，张强华主编. —6版. —北京：电子工业出版社，2014.5
ISBN 978-7-121-22119-4

Ⅰ. ①计… Ⅱ. ①司… ②张… Ⅲ. ①电子计算机-英语-高等学校-教材 Ⅳ. ①H31

中国版本图书馆CIP数据核字（2013）第296163号

策划编辑：秦　梅
责任编辑：桑　昀
印　　刷：
装　　订：
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路173信箱　邮编 100036
开　　本：787×1 092　1/16　印张：22　字数：736千字
印　　次：2014年5月第1次印刷
印　　数：4 000册　　定价：38.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

计算机科学是当今最具生命力的技术领域之一，其极高的发展速度、强劲的渗透能力、高附加值的经济价值使计算机技术进入了日新月异的发展时期。这也就决定了计算机领域中的新技术有着更短的生命周期，要求计算机行业的从业人员必须更快地掌握最新的技术。因此，计算机专业技术人员的英语水平比其他传统领域中同类人员的要求要高得多。可以毫不夸张地说，英语水平如何，是决定计算机技术人员成就大小的因素之一。

本书旨在切实提高读者实际应用英语的能力。本书立足实用，软件、硬件和网络并重，同时兼顾发展热点。

本书体例上以课为单元，每课由以下几部分组成：

1. 课文

这些课文选材广泛、风格多样。内容包括：个人计算机是如何工作的、可视显示单元、操作系统是如何工作的、数据结构、计算机程序、C 语言、关于 Java 技术、数据库基本概念、云计算、网络设备、软件工程、无线网络。

2. 单词

给出课文中出现的新单词，读者由此可以积累计算机专业的基本词汇。这些新词有的是《大学英语教学大纲》中没有而计算机专业中经常使用的单词，也有一些是《大学英语教学大纲》中有而计算机专业中有其他含义的单词。

3. 词组

给出课文中的常用词组。大部分是计算机专业词组，还有少量的计算机专业频繁使用的公共英语词组。

4. 难句讲解

讲解课文中出现的疑难句子，通过对其进行解剖和分析，帮助读者掌握分析、理解复杂句子的方法。

5. 语法

每课一个核心语法点，系统地讲述计算机领域中常见的语法。这些内容整合起来，可以作为“计算机英语简明语法手册”，对语法进行系统复习和学习。可以为读者的阅读、写作提供有力的支持。

6. 习题

既有语法练习，也有针对课文的练习，还提供了部分“计算机软件水平考试”真题（包括最近几年的程序员级、高级程序员级、系统分析师级试题）。另外，我们在每课中还安排了一道听力练习题，以提高读者的听力水平。

7. 技能训练

模拟了一个工作环境，以训练读者运用语言的能力。

8. 阅读材料

进一步扩大读者的视野，包括：嵌入式系统、打印机、操作系统、排序算法、网页设

计、面向对象编程、面向对象建模、数据挖掘、大数据、因特网一般用法、统一建模语言（UML）、物联网。

9. 参考译文

给出了课文的译文，供读者学习时对照、参考。

书后的“附录 A　英语基本句型”提供了常用的英语句型，以助读者在“汉译英”时心中有数；“附录 B　英语单词速记法”会有效地增加读者的词汇量，尤其是便于“破译”那些新构造出来的单词。本书特别配套了朗读资料，教师可以在多媒体教室播放、学生可以课后复听、自学者可以通过 MP3 播放机随时收听。朗读资料可登录电子工业出版社华信教育资源网（www. huaxin. edu. cn）免费下载。

我们认为“计算机英语”类教材必须进行“动态维护”。因此，在本次修订中，更换了部分内容，反映了最新的教学思路和技术进步，读者的一些建议在本次修订中也予以采纳。本书由司爱侠、张强华担任主编，吕淑文、张千帆和解煜晨参加了编写工作。

我们收到了许多教师的反馈意见，对其中的问题我们都及时给予了回答，希望大家不吝赐教。教师可以索取参考试卷、电子教案、词汇总表等资料。如果联系，请发电子邮件至：zqh3882355@ sina. com。

让我们共同努力，使本书成为一部“结构合理、取材得当、知识丰富、严谨大气”的优秀教材。

本书出版十余年来，经过多次修订、完善，已经得到广泛认可，被很多学校选为教材，不仅获得全国畅销书（科技类）、江苏省高教科研成果优秀奖，而且还被教育部批准为“普通高等教育‘十一五’国家级规划教材”。在此，对读者多年来的支持表示衷心感谢！

编　者

Contents

Lesson 1

Text

How PCs Work

When you mention the word "technology", most people think about computers. Virtually every facet of our lives has some computerized component. The appliances in our homes have microprocessors built into them, as do our televisions. Even our cars have a computer. But the computer that everyone thinks of first is typically the personal computer, or PC.

A PC is a general-purpose tool built around a microprocessor. It has lots of different parts—memory, a hard disk, a MODEM, etc. — that work together. "General-purpose" means that you can do many different things with a PC. You can use it to type documents, send E-mail, browse the Web and play games.

In this article, we will talk about PCs in the general sense and all the different parts that go into them. You will learn about the various components and how they work together in a basic operating session.

1. On the Inside

Let's take a look at the main components of a typical desktop computer.

- Central processing unit (CPU) — The microprocessor, "brain" of the computer system, is called the central processing unit. Everything that a computer does is overseen by the CPU.
- Memory — This is very fast storage used to hold data. It has to be fast because it connects directly to the microprocessor. There are several specific types of memory in a computer:
  - ⊙ Random-access memory (RAM) —Used to temporarily store information that the computer is currently working with.
  - ⊙ Read-only memory (ROM) — A permanent type of memory storage used by the computer for important data that does not change.
  - ⊙ Basic input/output system (BIOS) — A type of ROM that is used by the computer to establish basic communication when the computer is first turned on.
  - ⊙ Caching — The storing of frequently used data in extremely fast RAM that connects directly to the CPU.
  - ⊙ Virtual memory — Space on a hard disk used to temporarily store data and swap it in and out of RAM as needed.
- Motherboard—This is the main circuit board that all of the other internal components connect

to. The CPU and memory are usually on the motherboard. Other systems may be found directly on the motherboard or connected to it through a secondary connection. For example, a sound card can be built into the motherboard or connected through PCI (Peripheral Component Interconnect).

- Power supply—An electrical transformer regulates the electricity used by the computer.
- Hard disk—This is large-capacity permanent storage used to hold information such as programs and documents.
- Operating system—This is the basic software that allows the user to interface with the computer.
- Integrated Drive Electronics (IDE) Controller—This is the primary interface for the hard drive, CD-ROM (Compact Disc, Read-Only Memory) and floppy disk drive.
- PCI Bus—The most common way to connect additional components to the computer, PCI uses a series of slots on the motherboard that PCI cards plug into.
- Small Computer System Interface (SCSI) —Pronounced "scuzzy", the small computer system interface is a method of adding additional devices, such as hard drives or scanners, to the computer.
- Accelerated Graphics Port (AGP) —AGP is a very high-speed connection used by the graphics card to interface with the computer.
- Sound card—This is used by the computer to record and play audio by converting analog sound into digital information and back again.
- Graphics card—This translates image data from the computer into a format that can be displayed by the monitor.

2. Connections

2.1 Input/Output

No matter how powerful the components inside your computer are you need a way to interact with them. This interaction is called input/output (I/O). The most common types of I/O in PCs are:

- Monitor—The monitor is the primary device for displaying information from the computer.
- Keyboard—The keyboard is the primary device for entering information into the computer.
- Mouse—The mouse is the primary device for navigating and interacting with the computer.
- Removable storage devices—Removable storage devices allow you to add new information to your computer very easily, as well as save information that you want to carry to a different location.
  - ⊙ Floppy disk—The most common form of removable storage, floppy disks are extremely inexpensive and easy to save information to.
  - ⊙ CD-ROM—CD-ROM is a popular form of distribution of commercial software. Many systems now offer CD-R (recordable) and CD-RW (rewritable), which can also record.
  - ⊙ Flash memory—Based on a type of ROM called electrically erasable programmable read-

only memory (EEPROM). Flash memory provides fast, permanent storage. CompactFlash, SmartMedia and PCMCIA cards are all types of Flash memory.

⊙ DVD-ROM—DVD-ROM (digital versatile disc, read-only memory) is similar to CD-ROM but is capable of holding much more information.

2.2 Ports

- Parallel—This port is commonly used to connect a printer.
- Serial—This port is typically used to connect an external MODEM .
- Universal Serial Bus (USB) —Quickly becoming the most popular external connection, USB ports offer power and versatility and are incredibly easy to use.
- FireWire (IEEE 1394) —FireWire is a very popular method of connecting digital-video devices, such as camcorders or digital cameras, to your computer.

2.3 Internet/Network

- MODEM—This is the standard method of connecting to the Internet.
- Local area network (LAN) card—This is used by many computers, particularly those in an Ethernet office network, to connect each other.
- Cable MODEM—Some people now use the cable-television system in their home to connect to the Internet.
- Digital Subscriber Line (DSL) MODEM—This is a high-speed connection that works over a standard telephone line.
- Very high bit-rate DSL (VDSL) MODEM—A newer variation of DSL, VDSL requires that your phone line have fiber-optic cables.

3. From Power-up to Shut-down

3.1 BIOS

Now that you are familiar with the parts of a PC, let's see what happens in a typical computer session, from the moment you turn the computer on until you shut it down.

A. You press the "On" button on the computer and the monitor.

B. You see the BIOS software doing its thing, called the power-on self-test (POST). On many machines, the BIOS displays text describing such data as the amount of memory installed in your computer and the type of hard disk you have. During this boot sequence, the BIOS does a remarkable amount of work to get your computer ready to run.

- The BIOS determines whether the video card is operational. Most video cards have a miniature BIOS of their own that initializes the memory and graphics processor on the card. If they do not, there is usually video-driver information on another ROM on the motherboard that the BIOS can load.
- The BIOS checks to see if this is a cold boot or a reboot. It does this by checking the value at

memory address 0000:0472. A value of 1234h indicates a reboot, in which case the BIOS skips the rest of POST. Any other value is considered a cold boot.

- If it is a cold boot, the BIOS verifies RAM by performing a read/write test of each memory address. It checks for a keyboard and a mouse. It looks for a PCI bus and, if it finds one, checks all the PCI cards. If the BIOS finds any errors during the POST, it notifies you with a series of beeps or a text message displayed on the screen. An error at this point is almost always a hardware problem.
- The BIOS displays some details about your system. This typically includes information about the following:
  - ⊙ Processor;
  - ⊙ Floppy and hard drive;
  - ⊙ Memory;
  - ⊙ BIOS revision and date;
  - ⊙ Display.
- Any special drivers, such as the ones for SCSI adapters, are loaded from the adapter and the BIOS displays the information.

The BIOS looks at the sequence of storage devices identified as boot devices in the CMOS Setup. "Boot" is short for "bootstrap", as in the old phrase "Lift yourself up by your bootstraps". Boot refers to the process of launching the operating system. The BIOS tries to initiate the boot sequence from the first device using the bootstrap loader.

C. The bootstrap loader loads the operating system into memory and allows it to begin operation. It does this by setting up the divisions of memory that hold the operating system, user information and applications. The bootstrap loader then establishes the data structures that are used to communicate within and between the sub-systems and applications of the computer. Finally, it turns control of the computer over to the operating system.

3.2 Operating System

Once loaded, the operating system's tasks fall into six broad categories:

- Processor management—Breaking the tasks down into manageable chunks and prioritizing them before sending to the CPU.
- Memory management—Coordinating the flow of data in and out of RAM and determining when virtual memory is necessary.
- Device management—Providing an interface between each device connected to the computer, the CPU and applications.
- Storage management—Directing where data will be stored permanently on hard drives and other forms of storage.
- Application Interface—Providing a standard communications and data exchange between software programs and the computer.
- User Interface—Providing a way for you to communicate and interact with the computer.

You open up a word processing program and type a letter, save it and then print it out. Several components work together to make this happen:

- The keyboard and mouse send your input to the operating system.
- The operating system determines that the word-processing program is the active program and accepts your input as data for that program.
- The word-processing program determines the format that the data is in and, via the operating system, stores it temporarily in RAM.
- Each instruction from the word-processing program is sent by the operating system to the CPU. These instructions are intertwined with instructions from other programs that the operating system is overseeing before being sent to the CPU.
- All this time, the operating system is steadily providing display information to the graphics card, directing what will be displayed on the monitor.
- When you choose to save the letter, the word-processing program sends a request to the operating system, which then provides a standard window for selecting where you wish to save the information and what you want to call it. Once you have chosen the name and file path, the operating system directs the data from RAM to the appropriate storage device.
- You click on "Print". The word-processing program sends a request to the operating system, which translates the data into a format the printer understands and directs the data from RAM to the appropriate port for the printer you requested.

You open up a Web browser and check out "HowStuffWorks". Once again, the operating system coordinates all of the action. This time, though, the computer receives input from another source, the Internet, as well as from you. The operating system seamlessly integrates all incoming and outgoing information.

- You close the Web browser and choose the "Shut Down" option.
- The operating system closes all programs that are currently active. If a program has unsaved information, you are given an opportunity to save it before closing the program.
- The operating system writes its current settings to a special configuration file so that it will boot up next time with the same settings.

If the computer provides software control of power, then the operating system will completely turn off the computer when it finishes its own shut-down cycle. Otherwise, you will have to manually turn the power off.

New Words

technology	[tekˈnɔlədʒi]	*n.* 工艺，科技，技术
computer	[kəmˈpjuːtə]	*n.* 计算机，电脑
virtually	[ˈvəːtjuəli]	*adv.* 事实上，实际上，实质上

facet	[ˈfæsit]	*n*. 方面；（多面体的）面
computerize	[kəmˈpjuːtəraiz]	*vt*. 用计算机处理，使计算机化
component	[kəmˈpəunənt]	*n*. 部件
		adj. 组成的，构成的
appliance	[əˈplaiəns]	*n*. 设备，器械，装置
microprocessor	[maikrəuˈprəusesə]	*n*. 微处理器
oversee	[ˌəuvəˈsiː]	*vt*. 监督，监视；管理
memory	[ˈmeməri]	*n*. 存储器，内存
MODEM	[ˈməudəm]	*n*. 调制解调器
document	[ˈdɔkjumənt]	n. 文件，文档，公文
session	[ˈseʃən]	*n*. 对话期，运行期
type	[ˈtaip]	*n*. 类型，型，种类，样式
		v. 打字
E-mail	[ˈiːmeil]	*n*. 电子邮件
desktop	[ˈdesktɔp]	*adj*. 台式的，桌面的
		n. 桌面
storage	[ˈstɔridʒ]	*n*. 存储
data	[ˈdeitə]	*n*. 数据，资料
temporarily	[ˈtempərərili]	adj. 暂时的，临时的
store	[stɔː]	*vt*. 存储
information	[ˌinfəˈmeiʃən]	*n*. 信息
permanent	[ˈpəːmənənt]	*adj*. 永久的，持久的
establish	[isˈtæbliʃ]	*vt*. 建立，设立
communication	[kəˌmjuːniˈkeiʃn]	*n*. 通讯，通信
cache	[kæʃ]	*n*. 高速缓冲存储器
frequently	[ˈfriːkwəntli]	*adv*. 常常，频繁地，经常地
swap	[swɔp]	*v*. 交换
		n. 交换
motherboard	[ˈmʌðəbɔːd]	*n*. 主板，母板
transformer	[trænsˈfɔːmə]	*n*. 变压器
regulate	[ˈregjuleit]	*vt*. 管理，调整，控制，调节
capacity	[kəˈpæsiti]	*n*. 容量
program	[ˈprəugræm]	*n*. 程序
interface	[ˈintəfeis]	*n*. 界面，接口
software	[ˈsɔftwɛə]	*n*. 软件
controller	[kənˈtrəulə]	*n*. 控制器
bus	[bʌs]	*n*. 总线
plug	[plʌg]	*vt*. 插上
		n. 插头，插销

analog	[ˈænəlɔg]	*n.* 模拟
digital	[ˈdidʒitl]	*adj.* 数字的，数位的
		n. 数字，数字式
record	[ˈrekɔːd]	*n.* 纪录
		vt. 记录
audio	[ˈɔːdiəu]	*adj.* 音频的，声频的
format	[ˈfɔːmæt]	*n.* 格式
		vt. 格式化（磁盘）
display	[diˈsplei]	*vt.* 显示
		n. 显示，显示器
monitor	[ˈmɔnitə]	*n.* 监视器
		v. 监控
input	[ˈinput]	*n.* 输入
		v. 输入
output	[ˈautput]	*n.* 输出
		v. 输出
interaction	[ˌintərˈækʃən]	*n.* 交互作用
keyboard	[ˈkiːbɔːd]	*n.* 键盘
mouse	[maus]	*n.* 鼠标
removable	[riˈmuːvəbl]	*adj.* 可拆卸的，可移动的
location	[ləuˈkeiʃən]	*n.* 位置，场所
extremely	[iksˈtriːmli]	*adv.* 极端地，非常地
inexpensive	[ˌiniksˈpensiv]	*adj.* 便宜的，不贵重的
provide	[prəˈvaid]	*v.* 提供，供应，供给
parallel	[ˈpærəlel]	*adj.* 并行的
printer	[ˈprintə]	*n.* 打印机
port	[pɔːt]	*n.* 端口
serial	[ˈsiəriəl]	*adj.* 串行的，连续的
versatility	[ˌvəːsəˈtiləti]	*n.* 多功能性
camcorder	[ˈkæmkɔːdə]	*n.* 便携式摄像录音一体机
Internet	[ˈintənet]	*n.* 因特网，互联网
moment	[ˈməumənt]	*adj.* 片刻的，瞬间的
		n. 瞬间
boot	[buːt]	*v.* 导入，引导，启动
determine	[diˈtəːmin]	*v.* 决定，确定，测定
load	[ləud]	*n.* 负荷，加载
		vt. 装载，装入
check	[tʃek]	*vt.* 检查
address	[əˈdres]	*n.* 地址

		vt. 处理
indicate	[ˈindikeit]	*vt*. 指出，指明
beep	[biːp]	*n*. 蜂鸣声，哔哔声
		v. 嘟嘟响
message	[ˈmesidʒ]	*n*. 消息
detail	[ˈdiːteil]	*n*. 细节，详情
		vt. 详述，详解
hardware	[ˈhaːdwɛə]	*n*. 硬件
driver	[ˈdraivə]	*n*. 驱动器，驱动程序
category	[ˈkætigəri]	*n*. 种类，类目，部属，类别
instruction	[inˈstrʌkʃən]	*n*. 指令
save	[seiv]	*vt*. 保存
request	[riˈkwest]	*vt*. 请求，要求
		n. 请求，要求
click	[klik]	*v*. 单击，点击
browser	[brauzə]	*n*. 浏览器
receive	[riˈsiːv]	*vt*. 收到，接到，接收
option	[ˈɔpʃən]	*n*. 选项
active	[ˈækitiv]	*adj*. 活动的

Phrases

hard disk	硬盘
work with …	与……共事，与……合作，对……起作用
turn on	打开
virtual memory	虚拟内存
circuit board	电路板
connect to …	连接到……
power supply	电源
operating system	操作系统
floppy disk drive	软盘驱动器
sound card	声卡
as well as	也，又
base on …	基于……
digital camera	数码相机
fiber-optic cables	光缆
data exchange	数据交换
interact with…	与……交互，与……相互作用

shut down	关闭机器
turn off	关闭

Abbreviations

PC（Personal Computer）	个人计算机
CPU（Central Processing Unit）	中央处理器
RAM（Random-Access memory）	随机存储器
ROM（Read-Only Memory）	只读存储器
BIOS（Basic Input/Output System）	基本输入/输出系统
IDE（Integrated Drive Electronics）	电子集成驱动器
PCI（Peripheral Component Interconnect）	外部设备接口
SCSI（Small Computer System Interface）	小型计算机系统接口
CD-ROM（Compact Disc，Read-Only Memory）	只读光盘
EEPROM（Electrically Erasable Programmable Read-Only Memory）	电可擦除只读存储器
DVD-ROM（Digital Versatile Disc，Read-Only Memory）	只读数字化视频光盘
USB（Universal Serial Bus）	通用串行总线
LAN（Local Area Network）	局域网
DSL（Digital Subscriber Line）	数字用户线
VDSL（Very high bit-rate DSL）	超高速数字用户线
POST（Power-On Self-Test）	开机自检

Notes

[1] But the computer that everyone thinks of first is typically the personal computer，or PC.

本句中，that everyone thinks of first 是一个限定性定语从句，修饰和限定 the computer。that 在从句中作宾语，可以省略。or 引导同义词或说明语，意思是“或者说，即”。

[2] No matter how powerful the components inside your computer are you need a way to interact with them.

本句中，No matter how powerful the components inside your computer are 是一个让步状语从句，修饰谓语 need。

英语中，no matter 后面可以跟疑问代词 who、whom、which、what 或疑问副词 how、where，构成让步状语从句。请看下例：

No matter what you do you should put your heart into it.

无论做什么事情你都应该全心全意去做。

No matter how hard he tried he just couldn't solve the problem.

无论他多么努力，就是解决不了这个问题。

[3] Now that you are familiar with the parts of a PC，let's see what happens in a typical computer session，from the moment you turn the computer on until you shut it down.

本句中，Now that you are familiar with the parts of a PC 是一个原因状语从句，修饰谓语 let，from the moment you turn the computer on until you shut it down 对 a typical computer session 作进一步补充说明。

［4］On many machines, the BIOS displays text describing such data as the amount of memory installed in your computer and the type of hard disk you have.

本句中，describing such data as the amount of memory installed in your computer and the type of hard disk you have 是一个现在分词短语作定语，修饰和限定 text。在该短语中，installed in your computer 是一个过去分词短语，作定语修饰和限定 the amount of memory，you have 是一个定语从句，修饰和限定 the type of hard disk。

［5］The word-processing program sends a request to the operating system, which translates the data into a format the printer understands and directs the data from RAM to the appropriate port for the printer you requested.

本句中，which translates the data into a format the printer understands and directs the data from RAM to the appropriate port for the printer you requested 是一个非限定性定语从句，对 the operating system 作进一步补充说明。the printer understands 和 you requested 是定语从句，分别修饰和限定 a format 及 the appropriate port for the printer。

Grammar

定 语 从 句

在复合句中，修饰某一个名词或代词的从句称为定语从句。定语从句所修饰的名词或代词称为先行词。定语从句放在先行词的后面。请看下例：

【例】Do you know the man who will give us a talk on computer science tomorrow?

你认识明天要给我们作关于计算机科学的报告的那个人吗?

句中，who will give us a talk on computer science tomorrow 是定语从句，the man 是先行词。

【例】This is the software that I would like to buy.

这就是我想买的那个软件。

句中，that I would like to buy 是定语从句，the software 是先行词。

【例】He will never forget the day when he bought his own computer.

他永远都不会忘记自己买到计算机的那一天。

句中，when he bought his own computer 是定语从句，the day 是先行词。

通常，定语从句都由关系代词 that、which、who、whom、whose 和关系副词 when、where、why、how 引导。关系代词和关系副词往往放在先行词和定语从句之间，起联系作用，同时又作定语从句的一个成分。

定语从句可分为限定性定语从句和非限定性定语从句两类，以下分述之。

1. 限定性定语从句

限定性定语从句使修饰的词代表一个（些）或一类特定的人或物。如果修饰人，一般

用关系代词 who，有时也用 that。若关系代词在句子中作主语，则 who 用得较多且不可省略。请看下例：

【例】Those who agree with me please put up your hands.

同意我的观点的人请举手。

句中，who agree with me 是定语从句，修饰 Those。who 既是引导词，又在句中作主语，who 不能省略。

【例】Who is the man that is checking the printer over there?

在那边检查打印机的那个人是谁？

句中，that is checking the printer over there 是定语从句，修饰 the man。that 既是引导词，又在句中作主语，that 不能省略。

若关系代词在句子中作宾语，就应当使用宾格 whom 或 that，但在大多数情况下都可省略。若表示所属，就应用 whose。请看下例：

【例】He is the professor（whom）you've been looking for.

他就是你一直在寻找的教授。

句中，（whom）you've been looking for 是定语从句，修饰 the professor。whom 在从句中作 looking for 的宾语，故可省略。

【例】He is a man（that）you can depend on.

他是一个可以信赖的人。

句中，（that）you can depend on 是定语从句，修饰 a man。that 在从句中作 depend on 的宾语，故可省略。

【例】PCTOOLS are tools whose functions are very advanced.

PCTOOLS 是功能很先进的工具。

句中，whose functions are very advanced 是定语从句，修饰 tools。因为 functions 和 tools 之间是所属关系，所以用所有格 whose。

限定性定语从句如果修饰物，用 that 较多，也可用 which。它们可在句中作主语，也可作宾语。若作宾语，则大多可省略。请看下例：

【例】Some floppy disk drives that use 5.25-inch floppy disks have a lever that you need to push down or to the side after inserting the floppy disk.

一些用 5.25 英寸软盘的软盘驱动器有一个柄。在插入软盘之后，要把柄扳下来。

句中，that use 5.25-inch floppy disks 和 that you need to push down or to the side 是两个定语从句，分别修饰 some floppy disk drives 和 a lever，但第一个 that 不能省略，因为它在句中作主语。而第二个 that 则可省略，因为它在句中作 push 的宾语。

【例】Electricity is a form of energy which can be transmitted easily.

电是易输送的一种能量。

句中，which 引导的定语从句修饰 energy。因为 which 在从句中作主语，所以不能省略。

【例】Mouse is an instrument which operators often use.

鼠标是操作员经常使用的一种工具。

句中，which 引导的定语从句修饰 an instrument。因为 which 在从句中作 use 的宾语，所以可以省略。

当限定性定语从句所修饰的先行词是 that、all、only、everything、something、nothing 等代词，或者先行词前有一个最高级的形容词修饰时，则用 that 引导。但 that 若在从句中作宾语，则可省略。请看下例：

【例】It there anything I can do for you?

我能为你做些什么呢?

句中，先行词是 anything，故 that 省略。

【例】This is the most advanced computer we have ever seen.

这是我们所看过的最先进的计算机。

句中，先行词 computer 前面由最高级形容词 the most advanced 修饰，故 that 省略。

【例】AutoCAD is the best software that is found in CAD.

AutoCAD 是 CAD 领域中所找到的最好的软件。

句中，虽然由形容词的最高级 the best 修饰 software，但因 that 在定语从句中作主语，所以不能省略。

当限定性定语从句修饰一个表示时间的名词时，使用关系副词 when；当限定性定语从句修饰一个表示地点的名词时，使用 where；关系副词 why 的先行词通常是 reason。请看下例：

【例】The time will come when man can fly to outer space freely.

人类能自由地飞向外层空间的时代将会到来。

句中，先行词是 the time，故使用 when 引导。

【例】That's the place where we bought our printer.

那就是我们买打印机的地方。

句中，先行词 the place 是一个地点名词，故使用 where 引导。

【例】Do you know the reason why there are viruses in the computer?

你知道这台计算机中为什么会有病毒吗?

句中，先行词是 the reason，所以使用 why 引导。

2. 非限定性定语从句

非限定性定语从句一般用于对所修饰的名词或代词作进一步的说明，但也可对整个句子进行说明。它在修饰人时用 who、whom 或 whose，修饰物时用 which。请看下例：

【例】We do experiments with a computer, which helps to do many things.

我们利用计算机做实验，计算机可帮助做许多工作。

句中，which 引导的非限定性定语从句是对先行词 a computer 的说明。

【例】He arrived late, which was annoying.

他很迟才到，这真叫人恼火。

句中，which 引导的非限定性定语从句是修饰它前面的整个句子。

【例】Yesterday I met your Manager, who seemed to be very worried.

昨天我遇见你们经理，他好像很忧虑。

句中，who 引导的非限定性定语从句是对先行词 your Manager 的说明。非限定性定语从句也可由 where 或 when 引导。请看下例：

【例】They are going to buy a laser printer next week, when they will be free.

他们打算下周去买一台激光打印机，那时他们有空。

【例】They'll fly to America, where they plan to stay for a month.

他们将飞往美国，计划在那逗留一个月。

3. 限定性定语从句与非限定性定语从句的区别

限定性定语从句与主句的关系十分密切，它是整个句子必不可少的一个部分。如果把它去掉，句子的意思就不完整。而非限定性定语从句与主句的关系并不十分密切，它只是对所修饰的词或句子进行进一步说明，去掉之后句子的其他部分仍可成立。

限定性定语从句紧跟在它所修饰的词之后，而非限定性定语从句通常和句子的其他部分用逗号隔开。

在翻译中，限定性定语从句常译为定语，即“……的”；而非限定性定语从句常可译为一个并列的句子。

另外，非限定性定语从句不用 that 引导，这一点应加以注意。

4. 介词前置的定语从句

在口语中，若关系代词在从句中作介词的宾语时，通常用 whom、which 或 that 引导。此时，介词放在句子的后面，关系代词可以省略。但是，在正式书面语，特别是在科技英语中，介词放在关系代词之前。此时，只能使用关系代词 whom 或 which 引导，且不能省略关系代词。请看下例：

【例】Have you met the person about whom he was speaking?

你见过他说的那个人吗?

【例】A box within a dialog box in which you type information needed to carry out a command.

对话盒中的一个盒子，可以给该盒中输入执行一个命令所需的信息。

句中，in which…是一个介词前置的定语从句，修饰和限定 a box。

关系代词 whom 或 which 还可以作“动词词组”后面的介词的宾语。此时，应注意动词与介词的搭配。请看下例：

【例】The only thing about which he is not sure is how to use this tool.

他唯一没有把握的是如何使用这个工具。

句中的动词词组是 be sure about。

定语从句还可以由名词（代词） + of + which（whom）来引导，表示部分与整体的关系。注意不要误用 which 和 whom。请看下例：

【例】Our manager knows a lot of people, many of whom are professors.

我们经理认识许多人，其中好多是教授。

句中，whom 用来指人，具体就是指 people。

【例】She bought many books yesterday, all of which are on computer.

她昨天买了许多书，全是关于计算机方面的。

句中，which 指物，具体就是指 books。

Exercises

一、根据课文内容，判断以下叙述的正误。

（1） You can use PC to do a lot of things, such as type documents, send E-mail, browse the Web and play games.

（2） Memory is the brain of the computer.

（3） CPU oversees everything that a computer does.

（4） A sound card can only be built into the motherboard.

（5） Hard disks are used to hold information such as programs and documents.

（6） The monitor is the primary device for displaying information from the computer while the mouse is the primary device for navigating and interacting with the computer.

（7） A printer is usually connected into a serial port.

（8） An external MODEM is usually connected into a parallel port.

（9） Digital Subscriber Line（DSL）MODEM is a high-speed connection that works over a standard telephone line.

（10） Memory management is to coordinate the flow of data in and out of RAM and determine when virtual memory is necessary.

二、根据课文内容填空。

（1） CPU stands for ________________________.

（2） BIOS stands for ________________________.

（3） Motherboard is ____.

（4） Operating system is ____.

（5） Peripheral Component Interconnect（PCI）Bus is ______________________________.

（6） The keyboard is __________________________________.

（7） The most common form of removable storage is ________________________________.

（8） Many computers, particularly ________________________, use LAN card to ________________________.

（9） Device management is to ____.

（10） The operating system sends ________________________from the word-processing program to ________________________.

三、指出下列句子中的定语从句，然后把句子译成汉语。

（1） The room where we put our computers is very big.

（2） Users who are requiring the full color capabilities of the color VGA monitor will find that the color VGA monitor is a perfect choice.

（3） Do you know the reason why there are heat losses in a steam engine?

（4） The material which allows electric current to flow easily is called a conductor.

（5） Tom is the student whose father works in AAA Computer Company.

(6) They have invited us to visit their country, which is very kind of them.
(7) Iron is converted into steel by various processions, all of which involve heating it to very high temperatures.
(8) You should put the printer in a place where it is away from sunlight.
(9) The seventeenth century was one in which many significant advances were made in both science and philosophy.
(10) Galileo lived in the city of Pisa, where there is a leaning tower about 180 feet high.

四、选择与以下各条叙述意义最接近的词汇。

(1) One that is particularly concerned with the manipulation of files of numeric and non-numeric data and with the production of reports.
(2) One that is used mainly for the manipulation of numeric data.
(3) One that is designed to allow a programmer to make changes and corrections from a terminal during execution.
(4) One in which the user does not specify the sequences of operations that are to be performed to obtain a problem s solution.
(5) One that allows the programming of procedures that can be executed concurrently and can be activated in response to external signals as required.

供选择的答案：

A. interactive language
B. real-time language
C. scientific language
D. non-procedural language
E. commercial language

五、听句子，在画线处填写所听到的单词或词组。

(1) A personal computer (PC) is any ________ whose original sales price, size and capabilities make it useful for individuals.
(2) Today a PC may be a ________, a laptop computer or a tablet computer.
(3) Modern personal computers often have high-speed or dial-up connections to the Internet, allowing access to the ________ and a wide range of other resources.
(4) A PC may be a home computer, or may be found in an office, often connected to a ________.
(5) A desktop computer is an independent personal computer (PC), as opposed to smaller forms of PCs, such as a mobile ________.
(6) Desktop computers come in a variety of styles ranging from large vertical ________ to small form factor models that can be tucked behind an LCD monitor.
(7) Most modern desktop computers have separate screens and ________.
(8) A laptop computer or simply laptop, also called a notebook computer or sometimes a notebook, is a small personal computer ________ for mobility.

(9) Usually all of the interface hardware needed to operate the laptop, such as ________ and serial ports, graphics card, sound channel, etc., are built in to a single unit. Most laptops contain batteries to facilitate operation without a readily available electrical outlet.

(10) The ultra-mobile PC (UMPC) is a ________ for a small form factor tablet PC.

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

(1) A ________ is a functional unit that interprets and carries out instructions.

A. memory　　B. processor　　C. storage　　D. network

(2) A ________ consists of the symbols, characters, and usage rules that permit people to communicate with computer.

A. programming language　　B. network

C. keyboard　　D. display

(3) ________ software, also called end-user program, includes database programs, word processors, spreadsheets etc.

A. Application　　B. System　　C. Compiler　　D. Utility

(4) In ________, the only element that can be deleted or removed is the one that was inserted most recently.

A. a line　　B. a queue　　C. an array　　D. a stack

(5) Most ________ measures involve data encryption and password.

A. security　　B. hardware　　C. display　　D. program

Skill Training

个 人 简 历

简历（Resume）并没有固定不变的格式。应聘者应根据个人的具体情况、针对应聘企业的要求，选择恰当的形式，呈现适当的内容来设计、制作简历。

1. 简历的类型

一般来说，依据不同的侧重点，有以下三种类型。

1.1　以学历为主的简历（Basic Resume）

以学历为主的简历适应于应届毕业生，供其毕业求职使用。因为没有丰富的工作经历，所以把重点放在学业上，从最高学历写起。

1.2　以经历为主的简历（Chronological Resume）

以经历为主的简历往往侧重于工作经历。通常，根据应聘职位的要求，展现相关经历和业绩。一般按时间顺序书写出来，先写工作经历，再写学历。经历和学历的时间顺序均是由近及远。

1.3　以职能为主的简历（Functional Resume）

以职能为主的英语简历也是突出工作经历，因而所含元素与以经历为主的简历相同。两者的差别在于：以经历为主的简历是按时间顺序来排列工作经历，而以职能为主的简历则按工作职能或性质来概括工作经历，并无时间上的连贯性，旨在强调某些特定的工作能力和适应程度。例如，曾经在两个不同的工作单位担任相同的职务或负责相同的业务，便可归纳在一个项目之中。

2. 注意事项

2.1　充分了解招聘公司的要求

很多求职者都忽视了需要了解招聘公司情况，了解未来雇主的兴趣与问题。关于大多数公司的信息资料都是很丰富的，可以通过图书馆、杂志或因特网来搜集信息。关键在于要把自身情况与对方的要求相结合，根据对方需要来选择谈论你做过什么、能做什么，招聘者知道你能给公司带来何种效益。投放简历前一定要看清楚招聘方的要求，不要大撒网，既浪费钱，又容易打击自信心。不如根据自己的具体情况，精心选择，有的放矢地投递几家。

2.2　使简历醒目

在招聘现场，可以观察别人的简历都是用什么做封面包装，如果你的包装材料与众不同，就更容易被重视。

另外，简历封面的作用很大，招聘人员在拿起简历的时候，首先看到就是封面。设计精致或者别致的封面总会让人产生阅读的兴趣。在简历的包装上，要想让你的简历更为醒目显眼，最简单的方法就是做一个别致的简历封面。

对于电子简历而言，一般是通过邮件发送到对方邮箱，就没有封面的概念了，不过现在的邮箱都支持 html 格式，所以可以利用 html 格式设计表格、背景颜色、加粗字体，等等，这样比起纯文本格式的简历自然要好看得多，也方便阅读。发送简历的时候，一般不要以附件形式发送，因为接收者害怕附件中带有病毒。

2.3　有所保留

有一些事情最好还是留待面试时，而不是在简历中说明。原因很明显：这些事情可能会对求职者的录用带来不利。这些事情包括薪金要求、开始上班的时间，等等。你的目的是获得面试机会，如果他们真正认为你是合适的人选，那么在和他们面对面接触时，你可以提出这些求职中的棘手问题。

2.4　其他

注意以下问题：

（1）切忌简历中出现跳字、文字高低不平、用改正液涂改的痕迹。

（2）避免你所熟悉的缩写，招聘人员未必了解专业术语的缩写。

（3）照片可能让雇主对你产生错误的印象，如果对方没有要求，不要寄去。

（4）采用优质白纸，采用效果良好的打印机，如果你给的是复印件，效果要很好。如

果复印效果不理想，不要为了省钱将就着用复印的简历。一般来说，复印纸的质量也会比打印纸的质量差一些。

（5）在简历中要避免把不同纸型、不同纸质、不同颜色的纸张混杂在一起。

（6）如果打印出来后才发现错别字，不要犹豫，重新打印。

（7）最好不要采用简历模板，以免淹没在千篇一律的众多简历之中。

Reading Material

Embedded System

An embedded system is a computer system designed for specific control functions within a larger system, often with real-time[1] computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

The microprocessor embedded in this Adidas running shoe calculates the pressure between the runner's foot and the ground five million times per second and continuously changes the cushioning to match an adjustable comfort level.

Embedded systems contain processing cores that are either microcontrollers[2] or digital signal processors (DSP)[3].

A processor is an important unit in the embedded system hardware. It is the heart of the embedded system.

The key characteristic, however, is being dedicated to[4] handle a particular task. Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale[5].

Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary[6] installations like traffic lights, factory controllers, and largely complex systems like hybrid vehicles, MRI[7], and avionics[8]. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a

[1] real-time *adj.* 实时的

[2] microcontroller [ˈmaikrəʊkənˈtrəulə] *n.* 微型控制器

[3] digital signal processors (DSP) 数字信号处理器

[4] dedicate to 专注于

[5] economy of scale 规模经济

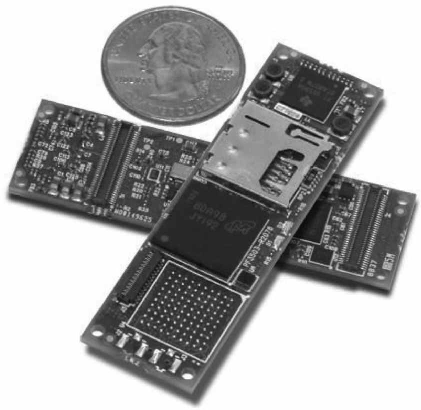
[6] stationary [ˈsteiʃənəri] *adj.* 固定的

[7] MRI (Magnetic Resonance Imaging) 核磁共振

[8] avionics [ˌeiviˈɔniks] *n.* 电子设备

large chassis or enclosure.

1. Variety of embedded systems



Gumstix Overo COM, a tiny, OMAP-based embedded computer-on-module with Wifi and Bluetooth.

Embedded systems are widespread in consumer, industrial, commercial and military applications.

Telecommunications[1] systems employ numerous embedded systems from telephone switches for the network to mobile phones[2] at the end-user. Computer networking uses dedicated routers and network bridges to route data.

Consumer electronics include personal digital assistants (PDAs)[3], mp3 players, mobile phones, videogame consoles[4], digital cameras, DVD players, GPS[5] receivers, and printers. Many household appliances[6], such as microwave ovens, washing machines and dishwashers[7], include embedded systems to provide flexibility, efficiency[8] and features. Advanced HVAC[9] systems use networked thermostats[10] to more accurately and efficiently control temperature that can change by time of day and season. Home automation uses wired and wireless-networking that can be used to control lights, climate, security, audio/visual, surveillance[11], etc., all of which use embedded devices for sensing and controlling.

Transportation systems from flight to automobiles increasingly use embedded systems. New airplanes contain advanced avionics such as inertial guidance systems[12] and GPS receivers that also have considerable safety requirements. Various electric motors—brushless DC motors[13], induction motors[14] and DC motors—use electric/electronic motor controllers. Automobiles, electric vehicles[15], and hybrid vehicles increasingly use embedded systems to maximize efficiency and reduce pollution. Other automotive safety systems include anti-lock braking system (ABS)[16],

[1] telecommunication [ˈtelikəmjuːniˈkeiʃən] *n.* 通讯，电信
[2] mobile phone 移动电话
[3] personal digital assistants (PDAs) 个人数字助理
[4] console [kənˈsəul] *n.* 控制台
[5] GPS (Global Position System) 全球定位系统
[6] household appliance 家用电器
[7] dishwasher [ˈdiʃˌwaʃə] *n.* 洗碗机
[8] efficiency [iˈfiʃənsi] *n.* 效率，功效
[9] HVAC (Heating, Ventilation and Air Conditioning) 供热通风与空气调节
[10] thermostat [ˈθəːməstæt] *n.* 自动调温器，温度调节装置
[11] surveillance [səːˈveiləns] *n.* 监视，监督
[12] inertial guidance system 惯性导引系统
[13] brushless DC motor 无刷直流电动机
[14] induction motor 感应电动机
[15] electric vehicle 电动车辆
[16] anti-lock braking system (ABS) 防抱死制动系统

Electronic Stability Control (ESC/ESP)[1], traction control system (TCS)[2] and automatic four-wheel drive[3].

Medical equipment uses embedded systems for vital signs monitoring, electronic stethoscopes for amplifying sounds, and various medical imaging (PET[4], SPECT[5], CT, MRI) for non-invasive internal inspections. Embedded systems within medical equipment are often powered by industrial computers.

Embedded systems are used in transportation, fire safety, safety and security, medical applications and life critical systems as these systems can be isolated from hacking and thus be more reliable. For fire safety, the systems can be designed to have greater ability to handle higher temperatures and continue to operate. In dealing with security, the embedded systems can be self-sufficient[6] and be able to deal with cut electrical and communication systems.

A new class of miniature wireless devices called motes[7] are quickly gaining popularity as the field of wireless sensor networking rises. Wireless sensor networking, WSN[8], makes use of miniaturization[9] made possible by advanced IC[10] design to couple full wireless subsystems to sophisticated sensors, enabling people and companies to measure a myriad of things in the physical world and act on this information through IT monitoring and control systems. These motes are completely self contained, and will typically run off a battery source for many years before the batteries need to be changed or charged[11].

Embedded Wi-Fi modules provide a simple means of wirelessly enabling any device which communicates via a serial port[12].

2. Characteristics

Embedded systems are designed to do some specific task, rather than be a general-purpose computer for multiple tasks. Some also have real-time performance constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified[13] to reduce costs.

Embedded systems are not always standalone devices. Many embedded systems consist of small, computerized parts within a larger device that serves a more general purpose. For example, the Gibson Robot Guitar features an embedded system for tuning the strings, but the overall purpose

---

[1] Electronic Stability Control (ESC/ESP) 电子稳定控制
[2] traction control system (TCS) 牵引控制系统
[3] four-wheel drive 四轮驱动
[4] PET (Position Emission Tomography) 正电子成像技术
[5] SPECT (Single-Photon Emission Computed Tomography) 单光子发射计算机断层扫描
[6] self-sufficient [ˈselfsəˈfiʃənt] *adj.* 自给自足的
[7] mote [məut] *n.* 尘埃，微粒
[8] WSN (Wireless Sensor Network) 无线传感器网络
[9] miniaturization [ˈminiətʃəraizəʃən] *n.* 小型化
[10] IC (Integrate Circuit) 集成电路
[11] charge [tʃaːdʒ] *n.* 电荷，充电
[12] serial port 串行端口
[13] simplify [ˈsimplifai] *vt.* 单一化，简单化

of the Robot Guitar is, of course, to play music. Similarly, an embedded system in an automobile provides a specific function as a subsystem of the car itself.

The program instructions written for embedded systems are referred to as firmware, and are stored in read-only memory or Flash memory[1] chips. They run with limited computer hardware resources: little memory, small or non-existent keyboard or screen.

2.1 Processors in embedded systems

Embedded processors can be broken into two broad categories. Ordinary microprocessors use separate integrated circuits for memory and peripherals. Microcontrollers have many more peripherals on chip, reducing power consumption[2], size and cost. In contrast to the personal computer market, many different basic CPU architectures are used, since software is custom-developed for an application and is not a commodity product[3] installed by the end user. Both Von Neumann as well as various degrees of Harvard architectures are used. RISC[4] as well as non-RISC processors are found. Word lengths vary from 4-bit to 64-bits and beyond, although the most typical remain 8/16-bit. Most architectures come in a large number of different variants and shapes, many of which are also manufactured by several different companies.

Numerous microcontrollers have been developed for embedded systems use. General-purpose microprocessors are also used in embedded systems, but generally require more support circuitry than microcontrollers.

2.2 ASIC[5] and FPGA[6] solutions

A common array of n configuration for very-high-volume embedded systems is the System on a Chip (SoC)[7] which contains a complete system consisting of multiple processors, multipliers, caches and interfaces on a single chip. SoCs can be implemented as an application-specific integrated circuit (ASIC) or using a field-programmable gate array (FPGA).

2.3 Peripherals

Embedded Systems talk with the outside world via peripherals, such as:

- Serial Communication Interfaces (SCI)[8]: RS-232, RS-422, RS-485 etc;
- Synchronous[9] Serial Communication Interface: I2C[10], SPI[11], SSC and ESSI (Enhanced

---

[1] Flash memory 闪存
[2] power consumption 能耗，功耗
[3] commodity product 商品
[4] RISC (Reduced Instruction Set Computing) 精简指令集
[5] ASIC (Application-Specific Integrated Circuit) 特定用途集成电路
[6] FPGA (Field-Programmable Gate Array) 现场可编程门阵列
[7] system on a chip (SoC) 集成的系统芯片
[8] Serial Communication Interface (SCI) 串行通信接口
[9] synchronous [ˈsiŋkrənəs] *adj.* 同时的，同步的
[10] I2C (Inter-Integrated Circuit) 内置集成电路
[11] SPI (Serial Peripheral Interface) 串行外部接口

Synchronous Serial Interface)；

- Universal Serial Bus (USB)；
- Multi Media Cards (SD Cards, Compact Flash etc.)；
- Networks: Ethernet, LonWorks[1], etc；
- Fieldbuses: CAN[2]-Bus, LIN[3]-Bus, PROFIBUS[4], etc；
- Timers: PLL[5] (s), Capture/Compare and Time Processing Units；
- Discrete IO: aka General Purpose Input/Output (GPIO)；
- Analog to Digital/Digital to Analog (ADC/DAC)；
- Debugging: JTAG[6], ISP[7], ICSP[8], BDM[9] Port, BITP[10] and DP[11] ports.

参 考 译 文

个人计算机是如何工作的

当提到“技术”这个词时，大多数人就会想到计算机。实际上，生活中的每一方面都有许多计算机化的成分。家庭中的许多用品（如电视机）都装有微处理器，甚至我们的汽车中也装有计算机。但是大家首先想到的计算机是个人计算机，即 PC。

PC 是围绕微处理器制成的常用工具。它由许多不同部件——内存、硬盘、调制解调器等——共同工作。“常用”意味着你可以使用 PC 做许多事情。你可以用它打印文稿、发送电子邮件、浏览网站和玩游戏。

本文将介绍一般意义上的 PC 及其各种部件。你将了解这些部件以及它们是如何协调工作的。

1. 内部部件

让我们看看典型的桌面计算机的主要部件。

- 中央处理器（CPU）——计算机系统的“大脑”，也被称为“中央处理器”。计算机做的每件事都是由 CPU 管理的。
- 内存——用来保存数据的快速存储设备。因为它直接与中央处理器相连所以运行速度很快。计算机中的内存有以下几种类型：

[1] LonWorks (local operation network Works) 局部操作网络
[2] CAN (Controller Area Network) 控制器区域网络
[3] LIN (Local Interconnect Network) 本地互联网络
[4] PROFIBUS (Process Field Bus) 过程现场总线
[5] PLL (Phase Lock Loop) 锁相回路，锁相环
[6] JTAG (Joint Test Action Group) 联合测试行动小组
[7] ISP (In-System Programming) 在系统编程
[8] ICSP (In Circuit Serial Programming) 在线串行编程
[9] BDM (Background Debug Mode) 背景调试模式
[10] BITP (Biotech Industrial Training Program) 生物技术工业训练编程
[11] DP (Display Port) 显示接口

⊙ 随机存储器（RAM）——用来临时存储计算机当前工作中的信息。

⊙ 只读存储器（ROM）——永久存储的存储器，计算机用来存储不改变的重要数据。

⊙ 基本输入/输出系统（BIOS）—— 一种 ROM，计算机用来建立首次开机时的基本通信。

⊙高速缓冲存储器 —— 把频繁使用的数据存储到直接与 CPU 连接的速度极快的 RAM 中。

⊙虚拟内存 —— 硬盘上的空间，用来临时存储数据，当需要时与 RAM 交换数据。

- 主板 —— 主要的电路板，其他所有内部部件都与其连接。CPU 和内存通常安装在主板上。其他系统可以直接安装在主板上或者通过附件连接到主板上。例如，声卡可以直接内置在主板上，也可以通过 PCI 连接到主板上。
- 电源 —— 一个电子变压器，调节计算机所用的电压。
- 硬盘 —— 这是大容量持久存储设备，用来保存像程序和文档这样的信息。
- 操作系统 —— 这是基础软件，是用户与计算机之间的接口。
- 电子集成驱动器（IDE）—— 这是用于硬盘、CD-ROM 和软盘驱动器的主要接口。
- 外部设备互连（PCI）总线 —— 连接计算机外设的最常用的通道。PCI 使用一系列位于主板上的 PCI 卡插槽。
- 小型计算机系统接口（SCSI）—— 音同“skuzzy”，这个小型计算机系统接口用于给计算机添加外围设备，如硬盘或扫描仪。
- 加速图形接口（AGP）—— AGP 是非常高速的连接接口，用于计算机和图形卡建立连接。
- 声卡 —— 用于记录和播放音频，计算机用它来把模拟音频信号转换为数字音频信号，也可以把数字音频信号再转换为模拟音频信号。
- 图形卡 —— 用来将计算机的图像数据转换为显示器可显示的格式。

2. 连接

2.1 输入/输出

无论计算机中的部件功能多么强大，都需要一种与它们交互的途径。这种交互途径称为输入/输出（I/O）。PC 中最常用的 I/O 类型有：

- 显示器 —— 显示器是用来显示计算机中信息的主要设备。
- 键盘 —— 键盘是把信息输入到计算机中的主要设备。
- 鼠标 —— 鼠标是用于浏览和与计算机交互的主要设备。
- 可移动存储设备 —— 可移动存储设备非常容易地向本地计算机中加入新的信息，也可以保存要带到其他地方的信息。

⊙ 软盘 —— 移动存储最常用的形式，软盘非常便宜而且方便保存信息。

⊙ 只读光盘（CD-ROM）—— CD-ROM 是商业软件发布最流行的形式。许多系统现在都提供 CD-R（可记录）和 CD-RW（可改写）设备，CD-RW 也可以记录。

⊙ 闪存 —— 基于电可擦除只读存储器（EEPROM）的一种 ROM。闪存提供快速的、可持久的存储。CompactFlash、SmartMedia 和 PCMCIA 都属于闪存。

⊙ 数字视频只读光盘（DVD-ROM）——DVD-ROM 与 CD-ROM 类似，但能存储更多的信息。

2.2 接口

- 并行接口——它通常用于连接打印机。
- 串行接口——它总是用来连接外置调制解调器。
- 通用串行总线（USB）——迅速成为最流行的外部连接设备，USB 接口供电且功能多样，并非常易于使用。
- 火线（IEEE 1394）——火线是把数字视频设备连接到计算机的一种非常流行的方法，例如便携式摄像录音一体机、数码相机等。

2.3 因特网/网络

- 调制解调器——这是连接到因特网的标准方法。
- 局域网（LAN）卡——这被许多计算机使用，特别是在一个以太办公网络中，用来互相连接。
- 线缆调制解调器——有些人使用家中的有线电视系统连接到因特网。
- 数字用户线（DSL）调制解调器——用于快速连接标准电话线。
- 超高速数字用户线（VDSL）调制解调器——DSL 较新的变种，VDSL 要求用户的电话线使用光缆。

3. 从开机到关机

3.1 基本输入/输出系统（BIOS）

既然已经熟悉了 PC 中的部件，就让我们看看在一个典型的计算机运行期间（从打开电源到关机）发生了什么。

A. 按下计算机和显示器上的“On”按钮。

B. BIOS 软件进行“开机自检”（POST）。对许多计算机来说，BIOS 显示文本信息，这些信息描述如计算机中安装内存的数量和硬盘类型这样的数据。在这个引导期中，BIOS 做许多重要的工作为计算机运行做好准备。

- BIOS 测定显卡是否可以使用。大多数显卡有自己的小 BIOS 来初始化显卡上的内存和处理器。如果显卡没有自己的 BIOS，那么 BIOS 可以装入主板上的另一个 ROM 中的显卡驱动信息。
- BIOS 检查是冷启动还是重新启动。通过查看内存地址 0000:0472 的值来做到这一点。值是 1234h 表明是重新启动。在这种情况下，BIOS 就跳过剩余的 POST。其他的值就表明是冷启动。
- 如果是冷启动，BIOS 通过对内存每一地址的读/写来校验 RAM。它检查键盘和鼠标，并且查找 PCI 总线，如果找到，就检查全部的 PCI 卡。如果在 POST 期间，BIOS 发现错误，就会发出一系列蜂鸣声或在屏幕上显示文本信息。这时出现的所有错误几乎都是硬件故障。

- BIOS 显示系统的一些详细信息。一般包括关于以下部件的信息：
  - ⊙ 处理器；
  - ⊙ 软盘和硬盘驱动器；
  - ⊙ 存储器；
  - ⊙ BIOS 版本和日期；
  - ⊙ 显示器。
- 任何特别的驱动程序，如用于 SCSI 适配器的驱动程序，都是从适配器装入的，BIOS 显示该信息。

BIOS 把指定的存储设备序列看作 CMOS Setup 中的引导设备。“Boot”是“bootstrap”的缩写，如老话所说的“自我提升”。引导指启动操作系统的过程。BIOS 试着从使用引导装入程序的第一个设备来启动引导序列。

C. 引导装入程序把操作系统装入内存并开始运行它。通过建立保存操作系统、用户信息和应用程序的分区来做到这一点。然后，引导装入程序建立用于在该计算机的子系统内及子系统间通信的数据结构。最后，由操作系统控制计算机的运行。

3.2 操作系统

一旦装入，操作系统的任务就可分成以下六大类：

- 处理器管理——把任务分解为可管理的块，并在发送到 CPU 之前排序。
- 内存管理——整理写入或读出 RAM 的数据流，并决定何时需要虚拟内存。
- 设备管理——提供连接到计算机、CPU 和应用程序的每个设备之间的接口。
- 存储管理——引导数据永久性地存储到硬盘或其他存储设备上。
- 应用接口——提供软件程序和计算机间标准的通信和数据交换。
- 用户界面——提供用户与计算机通信和相互作用的途径。

用户可以打开字处理程序并输入一封信，保存它，然后把它打印出来。要完成这一工作，需要以下几个部件协同工作：

- 键盘和鼠标把用户的输入发送给操作系统。
- 操作系统确定字处理程序已激活，并把用户的输入作为程序数据。
- 字处理程序确定所用的数据格式，并通过操作系统临时存储在 RAM 中。
- 来自字处理程序的每个指令都通过操作系统发送给 CPU。在发送到 CPU 之前，这些指令与受操作系统监控的其他程序的指令交织在一起。
- 这时，操作系统有规律地给图形卡提供显示信息，引导它们显示在显示器上。
- 当要保存这封信时，字处理程序发送一个请求给操作系统。然后出现一个标准窗口，用于选择保存和调用信息的地址和名称。一旦选择了文件名称和路径后，操作系统就把 RAM 中的数据引导到合适的存储设备中。
- 单击“Print”按键。字处理程序发送一个请求给操作系统，操作系统把数据转换为打印机可以理解的格式，并把 RAM 中的数据引导到所请求的打印机的适当端口。

打开一个网络浏览器并查看“HowStuffWorks”网站。操作系统再次调整它的全部行为。这时计算机就可以接收来自其他源的输入，如来自因特网的，或来自用户的。操作系统将无

缝地整合全部的输入/输出信息。

- 关闭网络浏览器并选择“Shut Down”选项。
- 操作系统将关闭当前活动的全部程序。如果一个程序有未保存的信息，在关闭该程序之前会得到一个保存它的提示。
- 操作系统将把当前的设置写到一个特殊的配置文件中，以便下次启动时使用相同的设置。

如果计算机提供了电源的软件控制，则操作系统在完成自己的关机操作过程后，会完全地关闭计算机。否则，需要手动关闭电源。

Lesson 2

Text

Visual Display Unit

A visual display unit, often called simply a monitor or display, is a piece of electrical equipment which displays images generated from the video output of devices such as computers without producing a permanent record. A newer monitor typically consists of a TFT (Thin-Film Transistor) LCD (Liquid Crystal Display), with most older monitors based around a cathode ray tube (CRT). The monitor comprises the display device, simple circuitry to generate and format a picture from video sent by the signals source, and usually an enclosure. Within the signal source, either as an integral section or a modular component, there is a display adapter to generate video in a format compatible with the monitor.

1. Screen size

The size of a display is typically given as the distance between two opposite screen corners. One problem with this method is that it does not distinguish between the aspect ratios of monitors with identical diagonal sizes, in spite of the fact that a shape of a given diagonal span's area decreases as it becomes less square. For example, a 4 : 3 21"monitor has an area of ~211 square inches, while a 16 : 9 21" widescreen has an area of only ~188 square inches.

This method of size measurement dates from the early days of CRT television when round picture tubes were in common use, which only had one dimension that described display size. When round tubes were used to display rectangular images, the diagonal measurement of these was equivalent to the round tube's diameter, hence this was used.

Another historically problematic practice is the direct measurement of a monitor's imaging element as its quoted size in publicity and advertising materials. Especially on CRT displays, a substantial portion of the imaging element is concealed behind the case's bezel.

2. Imaging technologies

As with television, many different hardware technologies exist for displaying computer-generated output:

- Liquid crystal display (LCD). TFT LCDs are the most popular display devices for new computers.
  - ⊙ Passive LCDs produce poor contrast, slow response, and other image defects. These were

used in most laptops until the mid 1990s.
   - ⊙ Thin Film Transistor LCDs give much better picture quality in several respects. Nearly all modern LCD monitors are TFT.
- Cathode ray tube (CRT).
   - ⊙ Raster scan computer monitors, which produce images using pixels. These were the most popular display device for older computers.
   - ⊙ Vector displays, as used on the Vectrex, many scientific and radar applications, and several early arcade machines (notably Asteroids) —always implemented using CRT displays due to requirement for a deflection system, though can be emulated on any raster-based display.
   - ⊙ Television sets were used by most early personal and home computers, connecting composite video to the television set using a modulator. Resolution and image quality were strongly limited by the display capabilities of television.
- Plasma Display Panel (PDP).
- Video projectors use CRT, LCD, DLP (Digital Light Processing), LCoS (Liquid Crystal on Silicon) or many other technologies to send light through the air to a projection screen. Front projectors use screens as reflectors to send light back, while rear projectors use screens as diffusers to refract light forward. Rear projectors are often integrated into the same case as their screen.
- Surface-conduction electron-emitter display (SED).
- Organic light-emitting diode (OLED) display.

2.1 Cathode ray tube (CRT)

The CRT or cathode ray tube, is the picture tube of a monitor. The back of the tube has a negatively charged cathode. The electron gun shoots electrons down the tube and onto a charged screen. The screen is coated with a pattern of phosphor dots that glow when struck by the electron stream. Each cluster of three dots, one of each color, is one pixel.

The image on the monitor's screen is usually made up of at least tens of thousands of such tiny dots glowing on command from the computer. The closer together the pixels are, the sharper the image on screen can be. The distance between pixels on a computer monitor screen is called its dot pitch and is measured in millimeters. Most monitors have a dot pitch of 0.28 mm or less.

There are two electromagnets around the collar of the tube which deflect the electron beam. The beam scans across the top of the monitor from left to right, is then blanked and moved back to the left-hand side slightly below the previous trace (on the next scan line), scans across the second line and so on until the bottom right of the screen is reached. The beam is again blanked, and moved back to the top left to start again. This process draws a complete picture, typically 50 to 100 times a second. The number of times in one second that the electron gun redraws the entire image is called the refresh rate and is measured in hertz (cycles per second). It is common in television or very early computer equipment to use a technique called interlacing, in which all the odd-numbered lines

of an image are traced, and then all the even-numbered lines; the circuitry of such an interlaced display need be capable of only half the speed of a non-interlaced display. An interlaced display, particularly at a relatively low refresh rate, can appear to some observers to flicker, and may cause eyestrain.

2.2 Liquid crystal display (LCD)

An LCD is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. It is often utilized in battery-powered electronic devices because it uses very small amounts of electric power.

3. Performance measurements

The performance parameters of a monitor are:

- Luminance, measured in candelas per square meter (cd/m^2).
- Viewable image size, measured diagonally. For CRTs the viewable size is typically one inch (25 mm) smaller than the tube itself.
- Display resolution, the number of distinct pixels in each dimension that can be displayed. Maximum resolution is limited by dot pitch.
- Dot pitch, describes the distance between pixels of the same color in millimetres. In general, the smaller the dot pitch (e.g., 0.24 mm), the sharper the picture will appear.
- Refresh rate, the number of times in a second that a display is illuminated. Maximum refresh rate is limited by response time.
- Response time, the amount of time a pixel in a monitor takes to go from active (black) to inactive (white) and back to active (black) again. It is measured in milliseconds (ms). Lower numbers mean faster transitions and therefore fewer visible image artifacts.
- Contrast ratio, the contrast ratio is defined as the ratio of the luminosity of the brightest color (white) to that of the darkest color (black) that the monitor is capable of producing.
- Power consumption, measured in watts (W).
- Aspect ratios, which is the horizontal size compared to the vertical size, e.g., 4 : 3 is the standard aspect ratio, so that a screen with a width of 1024 pixels will have a height of 768 pixels. A widescreen display can have an aspect ratio of 16 : 9, which means a display that is 1024 pixels wide will have a height of 576 pixels.
- Viewing angle, the ability to be seen from an angle without excessive degradation to the image, measured in degrees horizontally and vertically.

4. Comparison

4.1 CRT

Pros:

- Very high contrast ratio (20,000 : 1 or greater, much higher than many modern LCDs and

plasma displays).

- High speed response.
- Excellent color, wide gamut and low black level.
- Can display natively in almost any resolution and refresh rate.
- Near zero color, saturation, contrast or brightness distortion. Excellent viewing angle.
- No input lag.
- A reliable, proven display technology.

Cons:

- Large size and weight (a 40" unit weighs over 200lbs).
- Geometric distortion in non-flat CRTs.
- Older CRTs are prone to burn-in.
- Greater power consumption than same size LCD.
- Prone to moire effect at highest resolution.
- Intolerant of damp conditions.
- Small risk of implosion (due to internal vacuum) if the picture tube is broken.
- Lower refresh rates cause noticeable flicker.

4.2 LCD

Pros:

- Very compact and light.
- Low power consumption.
- No geometric distortion.
- Rugged.
- Little or no flicker depending on backlight.

Cons:

- Low contrast ratio.
- Limited viewing angle. This causes color, saturation, contrast and brightness to vary, even within the intended viewing angle from mere variations in posture.
- Uneven backlighting in some monitors can cause brightness distortion, especially toward the edges.
- Slow response times, which cause smearing and ghosting artifacts (although many modern LCDs have response times of 8ms or less).
- Only has one native resolution. Displaying other resolutions requires a video scaler, which degrades image quality at lower resolutions.
- Fixed bit depth, many cheaper LCDs are incapable of true color.
- Input lag.
- Somewhat more expensive than CRT.
- Dead pixels are possible during manufacturing.

4.3 PDP

Pros:

- Compact and light.
- High contrast ratios (10,000 : 1 or greater).
- High speed response.
- Excellent color, wide gamut and low black level.
- Near zero color, saturation, contrast or brightness distortion. Excellent viewing angle.
- No geometric distortion.
- Highly scalable, with less weight gain per increase in size (from less than 30 inches wide to the world's largest at 150 inches).

Cons:

- Large pixel pitch means either low resolution or a large screen.
- Noticeable flicker when viewed at close range.
- High operating temperature.
- Somewhat more expensive than LCD.
- High power consumption.
- Only has one native resolution. Displaying other resolutions requires a video scaler, which degrades image quality at lower resolutions.
- Fixed bit depth.
- Input lag.
- Older PDPs are prone to burn-in.
- Dead pixels are possible during manufacturing.

New Words

visual	[ˈvizjuəl]	*adj.* 看的，视觉的，形象的
equipment	[iˈkwipmənt]	*n.* 装备，设备，器材，装置
image	[ˈimidʒ]	*n.* 图像
video	[ˈvidiəu]	*n.* 视频
device	[diˈvais]	*n.* 设备，仪器，器件，装置
circuitry	[ˈsəːkitri]	*n.* 电路，线路
signal	[ˈsignəl]	*n.* 信号
		adj. 信号的
		v. 发信号，用信号通知
enclosure	[inˈkləuʒə]	*n.* 外壳；盒子，套
modular	[ˈmɔdjulə]	*adj.* 模块的
adapter	[əˈdæptə]	*n.* 适配器

compatible	[kəmˈpætəbl]	*adj.* 兼容的
screen	[skriːn]	*n.* 屏幕
distinguish	[disˈtiŋgwiʃ]	*v.* 区别，辨别
ratio	[ˈreiʃiəu]	*n.* 比，比率，比值
diagonal	[daiˈægənl]	*adj.* 对角线的，斜的
shape	[ʃeip]	*n.* 形状，外形，外观，样子
widescreen	[ˈwaidskriːn]	*n.* 宽屏幕
square	[skwεə]	*n.* 平方
measurement	[ˈmeʒəmənt]	*n.* 测量，度量
dimension	[diˈmenʃən]	*n.* 尺寸，尺度，维（数），度（数），元
element	[ˈelimənt]	*n.* 要素，元素，成分，元件
contrast	[ˈkɔntræst]	*n.* 对比度
response	[risˈpɔns]	*n.* 回答，响应，反应
defect	[ˈdiːfekt]	*n.* 过失，缺点，不足
raster	[ˈræstə]	*n.* 光栅
pixel	[ˈpiksəl]	*n.* 像素
vector	[ˈvektə]	*n.* 向量，矢量
radar	[ˈreidə]	*n.* 雷达
Asteroids	[ˈæstərɔids]	*n.* 行星战机（游戏机）
emulate	[ˈemjuleit]	*n.* 仿真
modulator	[ˈmɔdjuleitə]	*n.* 调制器，调节器
resolution	[ˌrezəˈljuːʃən]	*n.* 分辨率
plasma	[ˈplæzmə]	*n.* 离子体
reflector	[riˈflektə]	*n.* 反射体，反射镜
diffuser	[diˈfjuːzə]	*n.* 散射体
charged	[tʃɑːdʒd]	*adj.* 带电的
cathode	[ˈkæθəud]	*n.* 阴极，负极
tube	[ˈtjuːb]	*n.* 管，管子，[美] 电子管，显像管
phosphor	[ˈfɔsfə]	*n.* 磷光体，磷光剂
glow	[gləu]	*vi.* 发光，发热
cluster	[ˈklʌstə]	*n.* 串，束，群；类聚
electromagnet	[ilektrəuˈmægnit]	*n.* 电磁体，电磁铁
collar	[ˈkɔlə]	*n.* 颈，环管，轴环
redraw	[riːˈdrɔː]	*vi.* 刷新屏幕
		vt. 重画
hertz	[ˈhəːts]	*n.* 赫，赫兹（频率单位：周/秒）
interlacing	[ˌintə(ː)ˈleisiŋ]	*n.* 隔行扫描
flicker	[ˈflikə]	*vi.* 闪烁，抖动

eyestrain	[ˈaistrein]	*n.* 眼睛疲劳
monochrome	[ˈmɔnəukrəum]	*n.* 单色
		adj. 单色的
performance	[pəˈfɔːməns]	*n.* 性能
parameter	[pəˈræmitə]	*n.* 参数，参量
luminance	[ˈljuːminəns]	*n.* 亮度，辉度
candela	[kænˈdiːlə]	*n.* 烛光（发光强度单位）
sharp	[ʃɑːp]	*adj.* 锐利的，明显的，清晰的
illuminate	[iˈljuːmineit]	*vi.* 照亮，照射
luminosity	[ˌljuːmiˈnɔsiti]	*n.* 发光体，发光度
consumption	[kənˈsʌmpʃən]	*n.* 消费，消费量
degradation	[ˌdegrəˈdeiʃən]	*n.* 降级，降格，退化
excellent	[ˈeksələnt]	*adj.* 卓越的，极好的
gamut	[ˈgæmət]	*n.* 整个范围，全部；全音阶，音域
saturation	[ˌsætʃəˈreiʃən]	*n.* 饱和度
distortion	[disˈtɔːʃən]	*n.* 扭曲，变形，失真
lag	[læg]	*n.* 滞后，落后
reliable	[riˈlaiəbl]	*adj.* 可靠的，可信赖的
prove	[pruːv]	*vt.* 证明，证实，检验，考验
geometric	[dʒiəˈmetrik]	*adj.* 几何的，几何学的
burn-in	[ˈbəːn-in]	*n.* 老化
moire	[mwɑː]	*adj.* 波纹的
intolerant	[inˈtɔlərənt]	*adj.* 不宽容的；偏狭
damp	[dæmp]	*n.* 湿气
		adj. 潮湿的
implosion	[imˈpləuʒən]	*n.* 内爆，内破裂
vacuum	[ˈvækjuəm]	*n.* 真空
compact	[ˈkɔmpækt]	*adj.* 紧凑的，紧密的，简洁的
variation	[ˌvɛəriˈeiʃən]	*n.* 变更，变化，变异，变种
smearing	[ˈsmiəriŋ]	*n.* 拖尾效应
ghosting	[ˈgəustiŋ]	*n.* 重影

Phrases

a piece of	一块
consist of…	由……组成
signals source	信号源
in spite of	不管

arcade machine	游戏机（街机）
deflection system	偏转系统
negatively charged	带负电荷的
electron stream	电子束
be made up of…	由……组成
dot pitch	点距
refresh rate	刷新率
electric power	电能
viewable image size	可视图像尺寸
contrast ratio	对比度
power consumption	耗电量，能耗，功耗
aspect ratio	纵横比
viewing angle	视角

Abbreviations

TFT（Thin-Film Transistor）	薄膜晶体管
LCD（Liquid Crystal Display ）	液晶显示屏
CRT（cathode ray tube）	阴极射线管
DLP（Digital Light Processing）	数字光处理技术
LCoS（Liquid Crystal On Silicon）	硅基液晶（也缩写为 LCOS）
SED（Surface-conduction electron-emitter display）	表面传导电子发射显示
OLED（Organic light-emitting diode）	有机发光二极管
PDP（Plasma Display Panel）	等离子显示器

Notes

[1] A visual display unit, often called simply a monitor or display, is a piece of electrical equipment which displays images generated from the video output of devices such as computers without producing a permanent record.

本句中，often called simply a monitor or display 是一个过去分词短语，作主语 A visual display unit 的同位语，对其作补充说明。which displays images generated from the video output of devices such as computers without producing a permanent record 是一个定语从句，修饰和限定 a piece of electrical equipment。

[2] One problem with this method is that it does not distinguish between the aspect ratios of monitors with identical diagonal sizes, in spite of the fact that a shape of a given diagonal span's area decreases as it becomes less square.

本句中，that a shape of a given diagonal span's area decreases as it becomes less square 作 the fact 的同位语，说明其内容。distinguish 的意思是“区别，区分”，常与 from，between 连用。

in spite of 的意思是“不顾，不管”。请看下例：

Can you distinguish the different musical instruments playing now?

你能区分出正在演奏的各种不同的乐器吗?

Can you distinguish between those two objects?

你能区分那两个物体吗?

In spite of great efforts we failed to carry our plans through.

尽管我们做出了巨大努力，我们还是没能完成计划。

[3] The closer together the pixels are, the sharper the image on screen can be.

本句中，“the + 比较级，the + 比较级”的意思是“越……，越……”。请看下例：

The more angry he became, the more she laughed at him.

他越生气，她就越笑他。

The more difficult the questions are, the less likely I am to be able to answer them.

问题越困难，我就越不可能回答。

[4] It is common in television or very early computer equipment to use a technique called interlacing, in which all the odd-numbered lines of an image are traced, and then all the even-numbered lines; the circuitry of such an interlaced display need be capable of only half the speed of a non-interlaced display.

本句中，It 是形式主语，真正的主语是动词不定式 to use a technique called interlacing。in which all the odd-numbered lines of an image are traced, and then all the even-numbered lines 是一个非限定性定语从句，对 interlacing 作进一步补充说明，which 指 interlacing。be capable of 的意思是“能够，易于；有……的余地；可以，允许”。请看下例：

He is capable of judging art.

他具有鉴赏艺术的能力。

This tool machine is capable of being improved.

这台机床是可以改进的。

This is a room capable of 20 people.

这是一个可容纳 20 人的房间。

Grammar

状 语 从 句

在句中作状语的从句叫作状语从句。它用来修饰动词（包括谓语动词和非谓语动词）、形容词或副词。

状语从句可以放在句首，也可放在句末。放在句首时一般用逗号隔开，放在句末时一般不用逗号隔开。

状语从句通常由一个连词引起，有时也由一个起连词作用的词组引起，有时不用连词而直接和主句连接起来。请看下例：

【例】When you use FORMAT to format a disk that has never been formatted, specify the /u switch

to minimize formatting time.

当用 FORMAT 格式化一个从未格式化过的磁盘时，指定/u 开关可使格式化的时间减到最少。

句中，连词 When 引导了一个时间状语从句。

【例】Batteries should be kept in dry place for fear that electricity should leak away.

电池应置于干燥处，以免漏电。

句中，词组 for fear that 引导了一个目的状语从句。

【例】Say what you may, I won't change my mind.

不管你说什么，我都不会改变主意。

句中，Say what you may 是一让步状语从句。

状语从句可分为时间、地点、原因、目的、结果、条件、让步、比较和方式等九种。以下分述之。

1. 时间状语从句

表示时间的状语从句常常由 when、whenever、as、while、before、after、until、since、as soon as、once、as long as、now（that）、every time、the moment 等词引起。请看下例：

【例】After the jumper is set correctly, remove the cover of your computer and set it down someplace safe and out of the way.

正确地设置跳线卡之后，去掉计算机盖并把它放在不碍事的地方。

【例】When you choose a command that is followed by an ellipsis, a dialog box appears.

当选择一个后面跟省略号的命令时，对话框就会出现。

【例】Before you can work in an area in MS-DOS Shell, you need to select it.

在 MS-DOS Shell 一个区域中工作之前，应先选择它。

【例】We'll let you know as soon as it is arranged.

一等安排好我们就通知你。

【例】Once Scanning Gallery Plus 5.0 is installed, it can be opened from DOS or Windows.

一旦安装了 Scanning Gallery Plus 5.0 软件，就可以从 DOS 或 Windows 中打开它。

【例】He didn't leave his office until it was far into the night.

直到深夜他才离开了办公室。

【例】Each time I see him, he is at work.

每次我见到他，他都在工作。

【例】The machine will work the moment you press the button.

你一按按钮，机器就会启动。

2. 地点状语从句

表示地点的状语从句由 where 或 wherever 引导。请看下例：

【例】Where there is water, there is life.

哪里有水，哪里就有生命。

【例】Where there is a will, there is a way.

有志者事竟成。

【例】She found her book where she had left them.

她的书是在她原来放书的地方找到的。

【例】Please sit wherever you like.

请随便坐。

3. 原因状语从句

表示原因的状语从句可以由 as、because、since、seeing that、now that、considering that 引导。请看下例：

【例】The printer doesn't work because you press the wrong button.

因为按错了按钮，所以打印机不工作。

【例】I learn computer because I like it.

因为我喜欢计算机，所以我才学它。

【例】Since no one is against the plan, we'll adopt it.

既然没有人反对，这项计划就通过了。

【例】Now that everybody is here, let's discuss this design.

既然大家都来了，我们就讨论一下这个设计方案吧。

【例】Seeing that some of them were absent, they decided to put the meeting off.

由于一些人缺席，他们决定延期开会。

4. 目的状语从句

表示目的的状语从句通常由 so that、that、in order that、lest、for fear that、in case 等词引起。请看下例：

【例】We asked the professor to speak louder so that we could hear him.

我们请教授讲话声再大一些，以便让我们能听清。

【例】The manager wrote down the name of the new software for fear that he should forget it.

经理把新软件的名字记下来，以免忘了。

【例】In order that you can install the printer properly, you should read this manual.

要正确地安装打印机，必须先阅读这个手册。

【例】In order that people might understand his new invention, he made a practical demonstration of it.

为了让人们了解他的新发明，他做了一次示范表演。

5. 结果状语从句

结果状语从句可以由 that，so…that、so that、such…that 等词引导。请看下例：

【例】She didn't plan his time well, so that she didn't finish the work in time.

她没有把时间计划好，结果没有按时完成这项工作。

【例】This problem is so difficult that it will take us a lot of time to work it out.

这道题很难，我们要用很多时间才能解出。

【例】The current is so strong that the fuse often blows out.

电流过大使得保险丝经常烧断。

【例】They left in such a hurry that they forgot to lock the door.

他们走得太匆忙，连门都忘了锁。

【例】Some stars are so far away that we can't see them with our naked eyes.

有些恒星很遥远，用肉眼看不见。

6. 条件状语从句

条件状语从句由 if、unless、in case、so（as）long as、so far as、providing that、assuming that、provided that 引导。请看下例：

【例】If you don't want to give the disk a label, press ENTER.

如果不想标记磁盘，直接按 ENTER 键。

【例】Supposing we can't get the necessary equipment, what shall we do?

假定我们没有必要的设备，该怎么办呢？

【例】In case there is something wrong with the printer, it won't work properly.

倘若打印机出了故障，就不能正常运行。

【例】We'll go to buy some computers unless we are busy tomorrow.

如果我们明天不忙，就去买几台计算机。

【例】This is the most advanced software at home so far as I know.

据我所知，这是国内最先进的软件。

在虚拟条件句中，条件从句的前面有时可以不用连词。请看下例：

【例】Were I to do it, I could do it better.

要是我做这项工作，我会做得更好。

7. 让步状语从句

让步状语从句通常由 though、although、no matter、even if、even though、however、as、whatever、whether、however、whoever、whichever 等词引导。请看下例：

【例】Though time was short, they finished the work in time.

尽管时间很短，他们还是按时完成了这项工作。

【例】Your data is secure on your disk, no matter where you may leave your computer.

无论把计算机放在何处，磁盘上的数据都是安全的。

【例】Complicated as the problem is, it can be solved in an hour with the help of a computer.

虽然这个问题很复杂，但计算机在一小时内就可以解出。

【例】Much as computer languages differ, they have something in common.

尽管计算机语言之间各不相同，但它们仍有某些共同点。

【例】We'll carry on the work whether we can get the machines we need.

不管能否得到我们所需的机器，我们都要把这项工作继续下去。

【例】I won't believe you no matter what you say.

不论你说什么，我们都不会相信你。

【例】Even if he tried his best, he couldn't finish the work in time.
即使他竭尽全力，仍不能按时完成这项工作。

8. 比较状语从句

比较状语从句通常由 as、as…as、than、rather than、not so（as）…as 等词引导。由于里面常有一些成分没有表现出来，所以这类从句大都不完整。请看下例：

【例】He finished the work earlier than we had expected.
他完成这项工作比我们预计的要早。

【例】He doesn't work as hard as Tom.
他没有汤姆工作努力。

【例】The more I read this book, the more I like it.
这本书我越读越喜欢。

【例】The sooner, the better.
越快越好。

9. 方式状语从句

方式状语从句常用下列词引导：as though、as if、as、according as。请看下例：

【例】You can transfer files as if both computers' hard disks were in the same system.
你可以传输文件，就像两个计算机的硬盘在一个系统中一样。

【例】The earth itself behaves as though it were an enormous magnet.
地球本身的作用就像一个大磁铁一样。

【例】The manager told us to do as he told us.
经理告诉我们要按他说的去做。

【例】The thermometer rises or falls according as the air is hot or cold.
寒暑表随空气的冷热而升降。

【例】He acted as though nothing had happened.
他表现得就像什么也没有发生一样。

应该注意，在以 as if 及 as though 引导的方式状语从句中，谓语动词一般用虚拟语气。但是，如果方式状语从句中的情况实现的可能性较大时，也可用陈述语气。请看下例：

【例】It looks as if it is going to rain.
好像天要下雨。

Exercises

一、根据课文内容，判断以下叙述的正误。

（1）A visual display unit is often called simply a monitor or display.

（2）The size of a display is often measured by the distance between two opposite screen corners.

（3）CRTs are the most popular display devices for new computers.

（4）Most monitors have a dot pitch of 0.28 mm or more.

(5) For CRTs the viewable size is typically one inch (25 mm) smaller than the tube itself.
(6) In general, the bigger the dot pitch (e. g., 0.24 mm), the sharper the picture will appear.

二、根据课文内容填空。

(1) A newer monitor typically consists of a ________, with most older monitors based around a ____________________.
(2) The distance between pixels on a computer monitor screen is called ________________and is measured in ________.
(3) The number of times in one second that the electron gun redraws the entire image is called ____________________and is measured in ________.
(4) In ________, all the odd-numbered lines of an image are traced, and then all the ________ lines.
(5) A liquid crystal display (LCD) is often utilized in battery-powered electronic devices because ________________________________.
(6) Luminance is measured in ______________.
(7) Maximum refresh rate is limited by __________________.
(8) Response time is ___________________________a pixel in a monitor takes to go from active (black) to inactive (white) and back to active (black) again.
(9) Response time is measured in __________________. Lower numbers mean _________________________and therefore fewer visible image artifacts.
(10) Power consumption is measured in ________.

三、指出下列句子中的状语从句，并说明其在句中的作用，然后把句子译成汉语。

(1) If you are new to Microsoft Windows, this chapter is probably the most important one in the User's Guide since it explains the basic concepts and skills you need to work with Windows successfully.
(2) Whenever you see an ellipsis (…) after a menu command, a dialog box follows.
(3) Where there's smoke, there's fire.
(4) Wherever we go, we see electricity at work.
(5) Operating systems exist because they are supposed to make it easier to compute with an operating system than without it.
(6) The professor arrived while we were trying to phone him.
(7) Petrol has got too expensive, so we are selling the car.
(8) Late as it was, they still continued their work.
(9) Fewer people came to the meeting than I expected.
(10) I wish you'd stop talking so that we can hear what the others have to say.
(11) A good engineer learns as much from his mistakes as from his success.
(12) Now that you've come, let's discuss the design.

(13) Do as the manager told you.

(14) No matter how hard the work is, we'll try our best to finish it.

(15) You won't finish the work unless you hurry up.

四、选择与以下各条叙述意义最接近的词汇。

(1) the attribute of routine that allows the same copy of the routine to be used by two or more tasks.

(2) the address or the label of the instruction at which the computer program that called a subroutine is reentered from the subroutine.

(3) the attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

(4) the major application or designed procedure that equipment performs.

(5) a routine designed to find a hardware fault or to assist in diagnosing an error within a program.

(6) storage information remaining intact when power is turned off.

供选择的答案：

A. reentry point

B. malfunction routine

C. reusable

D. reentrant

E. permanent memory

F. main operation

五、听句子，在画线处填写所听到的单词或词组。

(1) A computer case is the enclosure that contains the main ________of a computer.

(2) A ________is the component that supplies power to a computer.

(3) Random-access memory (usually known by its acronym, RAM) is a type of computer data ________.

(4) A ________is a description of a class of logic machines that can execute computer programs.

(5) A ________ is the central or primary printed circuit board (PCB) making up a complex electronic system, such as a modern computer or laptop.

(6) Most motherboards produced today are designed for so-called IBM-________computers.

(7) A typical desktop computer is built with the ________, main memory, and other essential components on the motherboard.

(8) All of the basic circuitry and components required for a computer to ________are onboard the motherboard or are connected with a cable.

(9) The motherboard of a typical desktop ________a large printed circuit board.

(10) Motherboards contain some non-volatile memory to ________the system and load an operating system from some external peripheral device.

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

（1）Insufficient ________ can cause a processor to work at 50% or even more below its performance potential.

A. mouse　　B. I/O

C. document　　D. memory

（2）The ________ in E-mail messages has affected almost every computer around the world and has caused the damage of up to US $1 billion in North America.

A. illness　　B. virus

C. weakness　　D. attachment

（3）One of the basic rules of computer security is to change your ________ regularly.

A. name　　B. computer

C. device　　D. password

（4）One of the greatest features of a home ________ is the ability to share one Internet connection simultaneously over two or more computers.

A. computer　　B. device

C. network　　D. work

（5）The usual address for a Web site is the page address, although you can enter ________.

A. home　　B. main

C. host　　D. house

Skill Training

英语求职信

1. 格式

在英文求职信中，发件人的地址一般放在信的左上角或右上角，因此通常被称为“信头”。日期的位置也有讲究。缩进式的信，日期一般放在信头下的右上角；齐头式的信则放在左上角。在写法上，美式习惯为月、日、年排列，日期一般使用基数词，如“October 12, 2004”；按照英式习惯，日期的顺序为日、月、年，日期一般使用序数词，如“3rd November, 2004”。“年”前要有逗号，这点应该注意。

以下是齐头式英文求职信的范例：

Foreign Language Department
Fudan University
Shanghai　200052（发件人地址）

Oct. 10, 2004（日期）
China National Import & Export Corp.

No. 11 Fucheng Road,
Beijing（收件人地址）
Dear Sir or Madam:（称呼）
I am writing to inquire opportunities for…（应聘原因）
During the past four years…（说明能力）
Thank you for your consideration…（表示感谢）
Yours sincerely,（结尾用语）
Sun Hong（手写签名）
Encl.（附件）

2. 称呼用语

信函中使用的礼节性称呼很多。一般说来，给男士写信用“Mr.”（先生），给女士写信用“Miss”（小姐）、“Mrs.”（夫人）或“Ms.”（女士）。近年来，越来越多的人喜欢用“Ms.”来笼统地称呼女性，这样可以避免判断不清对方婚否等敏感话题所带来的尴尬，是一种大趋势。当然，也可以根据收信人的身份，用学位、头衔等常用词后加上收信人的姓，例如：Dr.（Doctor，博士）Zhang，Prof.（Professor，教授）Zhou，Pres.（President，经理、会长、校长）Liu，等等。如果对收信人的性别、姓名、职务或其他资料掌握不多的话，用一句“Dear sir or madam”最合适。如果收信人不是具体的某个人而是机关或者单位，则只写负责人的职位和公司或者机构的名称，例如：The Manager，CP Group 或 The President，Jinghui Company 等。

3. 正文

3.1 开头部分常用例句

My interest in the position of…has prompted me to forward my resume for your review and consideration.

因为对…… 一职感兴趣，所以奉上我的简历，请查阅和考虑。

Learning from Mr. Zhang that you are wanting a…I would like to apply to this position.

从张先生那里得知，贵单位正在招聘……人选，我愿意应聘此职。

The…position advertised in 51Job on October 12 intrigues me. I believe you will find me well-qualified.

我对 10 月 12 日《51Job》上刊登的…… 一职很感兴趣。相信您将发现我是这一职位的合适人选。

My desire to be one of your group prompted me to forward the attached for your consideration.

我想成为您团队中的一员，这种强烈愿望促使我向您奉上我的相关资料，希望您给予考虑。

I am writing to apply for the…position issued on your homepage. I believe my major and experiences will qualify me for the job.

根据贵公司网站主页上发布的招聘信息，给您写这封求职信，欲应聘……相信我的专业

和从业经验将能够胜任这份工作。

Having noticed the advertisement in today's BTV Channel 1, I wish to apply for the position referred to.

我想申请今天北京电视台 1 频道广告中所提及的职位。

3.2 自我推荐

I feel that I have the qualifications necessary to effectively handle the responsibilities of the position.

我认为自己具备高效率工作所必需的素质。

With more than two years of experience working as a part-time…in Beijing Sifang Technology Corp., I believe that I would be an excellent candidate of the position.

拥有在北京四方技术公司……方面长达两年的兼职经验，所以我相信自己是该职位出色的候选者。

I will finish my Bachelor's Degree in July at Fudan University. I am currently seeking full-time employment as an English teacher.

我将于今年 7 月完成复旦大学的本科学习，现在正在谋求全职英语教师一职。

I am presently a senior in good standing at Peking University, due to graduate in July receiving a Bachelor's Degree with a major in accounting. Enclosed please find the details and kindly for your consideration.

我现在是北京大学的优秀应届毕业生，即将在 7 月份毕业并获得会计专业的学士学位。详情请见附件，谢谢考虑。

For the past three years, I have achieved satisfactory scores in my studies and been awarded scholarships several times. I have a good command of computer, as well as English, due to my industriousness.

过去的三年里，我在学业上取得了令人满意的成绩，获得过多次奖学金。由于勤奋好学，在学好英语的同时，计算机能力也不错。

I took an active part in various activities concerning my major. These experiences contributed to the expansion of my knowledge scope and greatly improved my abilities.

我积极地参加了各种与我专业相关的活动。这些经历使我扩大了知识面，提升了我的能力。

3.3 结尾

The enclosed resume explicates my qualifications for the position advertised. I expect the opportunity to personally discuss my qualifications with you at your convenience.

附件中的简历详细说明了我具备招聘广告中所要求的素质。我期待着在您方便时有机会与您面谈。

Thank you for your attention to my resume. I welcome the opportunity to meet with you to further discuss my qualifications and your needs.

感谢您对我的关注。希望有机会跟您进一步讨论我的资历和您的要求。

I have enclosed a resume as well as a photo of my recent woks for your review. I look forward to an interview to discuss further how I could contribute to your organization.

随信奉上我的简历和近期作品照片一张，以供参考。期待您给我面试机会，以进一步探讨我能为贵公司做些什么。

I would appreciate your reply. Please contact me at 13612345678 any time, thanks for your time and consideration.

如蒙回复，不胜感激。请致电 13612345678，任何时候均可。感谢您拨冗考虑。

4. 结束用语

结束用语是写在结尾的告别话语，也称结尾问候语。位于正文下面隔两三行的位置，一般从中间偏右的地方开始写起。常用的结束用语主要有：Sincerely、Yours sincerely、Yours truthful、Yours faithfully、Yours truly、Yours 等，首字母大写而且每个词后都要加逗号。

Reading Material

Printer

A computer printer, or more commonly a printer, is a computer peripheral which produces a hard copy[1] (permanent human-readable text and/or graphics) of documents stored in electronic form, usually on physical print media such as paper or transparencies[2]. Many printers are primarily used as local peripherals, and are attached by a printer cable to a computer which serves as a document source. Some printers, commonly known as network printers, have built-in network interfaces (typically wireless or Ethernet), and can serve as a hardcopy device for any user on the network. Individual printers are often designed to support both local and network connected users at the same time.

In addition, a few modern printers can directly interface to electronic media such as memory sticks[3] or memory cards[4], or to image capture devices such as digital cameras, scanners; some printers are combined with a scanners and/or fax machines in a single unit. Printers that include non-printing features are sometimes called Multi-Function[5] Printers (MFP) or Multi-Function Devices (MFD).

A printer which is combined with a scanner can function as a kind of photocopier[6] if so designed. Most MFPs include printing, scanning, and copying among their features. Printers are designed for low-volume[7], short-turnaround print jobs; requiring virtually no setup time to achieve a hard copy of a given document. However, printers are generally slow devices (30 pages per minute

[1] hard copy 硬拷贝
[2] transparency [træns'pεərənsi] *n.* 幻灯片
[3] memory stick 内存条
[4] memory card 内存卡
[5] multi-function [ˌmʌlti'fʌŋkʃən] *n.* 多功能
[6] photocopier ['fəutəukɔpiə] *n.* 复印机
[7] low-volume 少量

is considered fast; and many consumer printers are far slower than that), and the cost-per-page is relatively high. In contrast, the printing press[1], which serves much the same function, is designed and optimized for high-volume[2] print jobs such as newspaper print runs. Printing presses are capable of hundreds of pages per minute or more, and have an incremental cost-per-page which is a fraction of that of printers.

The printing press remains the machine of choice for high-volume, professional publishing. However, as printers have improved in quality and performance, many jobs which used to be done by professional print shops are now done by users on local printers; see desktop publishing[3]. The world's first computer printer was a 19th century mechanically driven apparatus invented by Charles Babbage for his Difference Engine[4].

1. Printing technology

Printers are routinely classified by the underlying print technology they employ; numerous such technologies have been developed over the years.

The choice of print engine has a substantial[5] effect on what jobs a printer is suitable for, as different technologies are capable of different levels of image/text quality, print speed, low cost, noise[6]; in addition, some technologies are inappropriate[7] for certain types of physical media (such as carbon paper or transparencies).

Another aspect of printer technology that is often forgotten is resistance to alteration: liquid ink such as from an inkjet head[8] or fabric ribbon[9] becomes absorbed by the paper fibers, so documents printed with a liquid ink sublimation printer are more difficult to alter than documents printed with toner[10] or solid inks, which do not penetrate[11] below the paper surface.

Checks should either be printed with liquid ink[12] or on special "check paper with toner anchorage". For similar reasons carbon[13] film ribbons for IBM Selectric typewriters bore labels warning against using them to type negotiable instruments[14] such as checks. The machine-readable lower portion of a check, however, must be printed using MICR[15] toner or ink. Banks and other clearing houses employ automation equipment that relies on the magnetic flux from these specially

---

[1] press [pres] *n.* 印刷
[2] high-volume 大量
[3] desktop publishing 桌面出版
[4] Difference Engine 差分机
[5] substantial [səbˈstænʃəl] *adj.* 实际的，真实的
[6] noise [nɔiz] *n.* 噪声，噪音
[7] inappropriate [ˌinəˈprəupriit] *adj.* 不适当的，不相称的
[8] inkjet head 喷（墨）头
[9] fabric ribbon 色带
[10] toner [ˈtəunə] *n.* 调色剂，碳粉
[11] penetrate [ˈpenitreit] *vt.* 穿透，渗透
[12] ink [iŋk] *n.* 墨水
[13] carbon [ˈkɑːbən] *n.* 碳；（一张）复写纸
[14] instrument [ˈinstrumənt] *n.* 工具，手段，器械，器具
[15] MICR (Magnetic Ink Character Recognition) 磁墨水字符识别

printed characters to function properly.

2. Modern print technology

The following printing technologies are routinely found in modern printers:

2.1 Toner-based printers

Toner-based printers work using the Xerographic principle that is at work in most photocopiers: by adhering toner to a light-sensitive[1] print drum, then using static electricity to transfer the toner to the printing medium to which it is fused with heat and pressure.

The most common type of toner-based printer is the laser printer[2], which uses precision lasers to cause adherence[3]. Laser printers are known for high quality prints, good print speed, and a low (Black and White) cost-per-copy; they are the most common printer for many general-purpose office applications. They are far less commonly used as consumer printers due to a high initial cost.

Laser printers are available in both color and monochrome varieties[4].

Another toner based printer is the LED[5] printer which uses an array of LEDs instead of a laser to cause toner adhesion to the print drum.

Recent research has also indicated that laser printers emit potentially dangerous ultrafine[6] particles[7], possibly causing health problems associated with respiration[8] and cause pollution[9] equivalent to cigarettes. The degree of particle emissions varies with age, model and design of each printer but is generally proportional to the amount of toner required. Furthermore, a well ventilated workspace would allow such ultrafine particles to disperse thus reducing the health side effects.

2.2 Liquid inkjet printers[10]

Inkjet printers operate by propelling variably-sized droplets[11] of liquid or molten[12] material (ink) onto almost any medium. They are the most common type of computer printer for the general consumer due to their low cost, high quality of output, capability of printing in vivid[13] color, and ease of use.

---

[1] light-sensitive 光敏
[2] laser printer 激光打印机
[3] adherence [əd'hiərəns] *n.* 黏附，胶着
[4] variety [və'raiəti] *n.* 品种，种类
[5] LED (Light-Emitting Diode) 发光二极管
[6] ultrafine [ˌʌltrə'fain] *adj.* 极其细小的，非常细微的
[7] particle ['pɑːtikl] *n.* 微粒；粒子
[8] respiration [ˌrespi'reiʃən] *n.* 呼吸
[9] pollution [pə'luːʃən] *n.* 污染
[10] inkjet printer 喷墨打印机
[11] droplet ['drɔplit] *n.* 小滴
[12] molten ['məultən] *adj.* 熔化的，熔解的
[13] vivid ['vivid] *adj.* 鲜艳的，清晰的，逼真的

Like most modern technologies, the present-day inkjet has built on the progress made by many earlier versions. Among many contributors, Epson, Hewlett-Packard and Canon can claim a substantial share of the credit for the development of the modern inkjet. In the worldwide consumer market, four manufacturers account for the majority of inkjet printer sales: Canon, Hewlett-Packard, Epson and Lexmark.

The emerging ink jet material deposition market also uses ink jet technologies, typically piezoelectric[1] jets, to deposit materials directly on substrates.

2.3 Solid ink printers

Solid Ink printers, also known as phase-change printers, are a type of thermal transfer printer[2]. They use solid sticks of CMYK[3] (short for Cyan, Magenta, Yellow and Black), colored ink (similar in consistency to candle wax), which are melted and fed into a piezo crystal[4] operated printhead. The printhead[5] sprays[6] the ink on a rotating, oil coated drum. The paper then passes over the print drum, at which time the image is transferred or transfixed[7], to the page.

Solid ink printers are most commonly used as color office printers, and are excellent at printing on transparencies and other non-porous[8] media. Solid ink printers can produce excellent results. Acquisition and operating costs are similar to laser printers. Drawbacks of the technology include high power consumption and long warm-up times[9] from a cold state.

Also, some users complain[10] that the resulting prints are difficult to write on (the wax tends to repel inks from pens), and are difficult to feed through Automatic Document Feeders[11], but these traits have been significantly reduced in later models. In addition, this type of printer is only available from one manufacturer, Xerox, manufactured as part of their Xerox Phaser office printer line. Previously, solid ink printers were manufactured by Tektronix, but Tek sold the printing business to Xerox in 2001.

2.4 Dye-sublimation printers[12]

A dye-sublimation printer (or dye-sub printer) is a printer which employs a printing process that uses heat to transfer dye to a medium such as a plastic card, paper or canvas[13]. The process is

---

[1] piezoelectric [paiˌiːzəui'lektrik] *adj.* 压电的
[2] thermal transfer printer 热转印打印机
[3] CMYK：用于印刷的四分色（Cyan 青，Magenta 品红，Yellow 黄，blacK 黑）
[4] crystal ['kristl] *adj.* 结晶状的
n. 晶体
[5] printhead ['printhed] *n.* 打印头
[6] spray [sprei] *vt.* 喷射，喷溅
[7] transfix [træns'fiks] *vt.* 使固定
[8] non-porous 不吸水的
[9] warm-up times 预热时间
[10] complain [kəm'plein] *vi.* 抱怨，埋怨
[11] Feeder ['fiːdə] *n.* 送纸器，进纸器
[12] dye-sublimation printer 热升华打印机
[13] canvas ['kænvəs] *n.* 帆布

usually to lay one color at a time using a ribbon that has color panels. Dye-sub printers are intended primarily for high-quality color applications, including color photography; and are less well-suited for text. While once the province of high-end print shops, dye-sublimation printers are now increasingly used as dedicated consumer photo printers.

2.5 Inkless printers[1]

Inkless printers use paper with colorless dye crystals embedded between the two outer layers of the paper. When the printer is turned on, heat from the drum causes the crystals to colorize at different rates and become visible. One inkless printing technology, Zink, originally developed at Polaroid, became available in 2007. Because of the way it prints, the printer can be as small as a business card, the images are waterproof[2], and in fact, one product slated for release by Zink Imaging is a digital camera with a printer built into it.

Xerox is also working on an inkless printer which will use a special reusable paper coated with a few micrometres of UV[3] light sensitive chemicals. The printer will use a special UV light bar which will be able to write and erase the paper. As of early 2007 this technology is still in development and the text on the printed pages can only last between 16 ~ 24 hours before fading[4].

参 考 译 文

可视显示单元

可视显示单元通常简单地被称为监视器或显示器，它是一个电子设备，用来显示由视频输出设备（如计算机）产生的图像，但不会产生一个持久的记录。较新的监视器通常由薄膜晶体管液晶显示屏（TFT LCD）组成，而大多数老式的监视器都是基于阴极射线管（CRT）。监视器包含显示设备、产生和格式化由信号源发送的来自视频图像的简单电路，通常还有一个外壳。在信号源中，无论是作为集成部分还是模块部件，都有一个显示适配器来产生与监视器格式兼容的视频。

1. 屏幕尺寸

通常显示器尺寸就是屏幕两对角之间的距离。这种度量方式的一个问题是它不能区分显示器的外观比与同样的对角尺寸，尽管当越不接近正方形时其对角方向的面积越小。例如，一个 4∶3 的 21 英寸监视器的面积大约是 211 平方英寸，而一个 16∶9 的 21 英寸宽屏监视器的面积大约只有 188 平方英寸。

这种尺寸度量方法源于早期的 CRT 电视，当时普遍使用圆形显像管，只有一维来描述

[1] Inkless printer 无墨水打印机
[2] waterproof [ˈwɔːtəpruːf] *adj.* 防水的，不透水的
[3] UV (ultraviolet) 紫外光，紫外辐射
[4] fade [ˈfeid] *n.* 褪色，枯萎，衰退

显示尺寸。当用圆形显像管来显示矩形图像时，其对角线的度量就等于圆形显像管的直径，从此就使用了这种方法。

历史上另外一个值得怀疑的方法是直接度量显示器的图像元素，将其作为公开的和广告材料中的尺寸。特别是对 CRT 显示器，实际上一部分图像元素隐藏在机壳边框之下。

2. 成像技术

与电视一样，许多不同的硬件技术用于显示计算机的输出：

- 液晶显示器（LCD）。TFT LCD 是用于新型计算机的最流行的显示设备。
  - ⊙ 被动式 LCD 有对比度差、响应慢及其他图像缺点。20 世纪 90 年代中期以前主要用在便携式计算机上。
  - ⊙ 薄膜晶体管 LCD 在几个方面都有更好的图形质量。几乎现代所有的 LCD 都是 TFT 的。
- 阴极射线管（CRT）。
  - ⊙ 光栅扫描计算机显示器，使用像素产生图像。这些都是老式计算机最常用的显示设备。
  - ⊙ 向量显示，用在 Vectrex、许多科学和雷达应用设备以及早期游戏机（最典型的是 Asteroids）上，——总是使用 CRT 显示，原因在于需要一个偏转系统，虽然可以在任何基于光栅的显示器上来模拟。
  - ⊙ 许多早期的个人和家庭计算机也使用电视机，用调制器把复合视频连接到电视机上。电视机的分辨率和图像质量都是非常有限的。
- 等离子显示器。
- 视频投影仪使用 CRT、LCD、DLP、LCoS 或其他许多技术把光穿过空气投射到屏幕上。前面的投影仪把屏幕作为返回光线的反射器，而后面的投影仪把屏幕作为散射器来折射前面的光线。后面的投影仪通常与显示屏整合到同一机箱中。
- 表面传导电子发射显示（SED）。
- 有机发光二极管（OLED）显示。

2.1 阴极射线管

阴极射线管（CRT）是监视器的显像管。该管的后面有一个带负电的阴极。电子枪沿着射线管把电子发射到带电的屏幕上。屏幕涂了一层磷点，这些磷点在电子束的撞击下可以发光。每个颜色一点，三个点为一组，一组为一个像素。

监视器上的图像通常由至少数万这样的小点组成，按计算机命令来发光。像素的距离越近，屏幕上的图像越清晰。计算机显示器屏幕上像素之间的距离称为点距，以毫米来度量。大多数监视器的点距是 0. 28 毫米或更小。

在射线管管颈周围有两个偏转电子束的电磁体。这个电子束从显示器的顶部由左向右扫描、然后消失并移动到左边稍微下一点的位置（下一个扫描行），扫描第二行，如此扫描，直到屏幕的最后一行的右边。电子束消失，然后移回左上角再次开始。这个过程就画出了一个完整的图片，通常每秒 50 到 100 次。电子枪每秒刷新整个图像的次数称为刷新率，以赫兹来度量（每秒循环数）。在电视机和很早的计算机中普遍使用隔行扫描技术，该技术先扫

描图像的所有奇数行，然后再扫描图像的所有偶数行。这样一个隔行扫描显示器的电路所需的速度只是一个非隔行扫描显示器的1/2。对一些人来说，一个隔行扫描显示器，尤其是刷新率较低的隔行扫描显示器，看上去会有些抖动，并且会产生眼疲劳。

2.2 液晶显示器

液晶显示器（LCD）是一个薄的平板显示设备，由排列在光源或反射体前的任意色彩数或单色像素组成。由于液晶显示器耗电量很少，因此经常用在电池驱动的电子设备中。

3. 性能度量

监视器的性能参数是：

- 亮度，以每平方米坎［德拉］度量（坎［德拉］/平方米）。
- 可视图像大小，对角测量。对 CRT 而言，可视尺寸通常比其显像管小 1 英寸（25mm）。
- 显示器分辨率，每一维可以显示的确切像素数。最大分辨率受点距限制。
- 点距，以毫米描述同一颜色像素的距离。通常，点距越小（如0.24mm）显示的图像越清晰。
- 刷新率，每秒照亮显示器的次数。最高刷新率受响应时间限制。
- 响应时间，显示器中像素从激活（黑）到非激活（白）再到激活所需要时间的值，以毫秒度量。数值越小，转换越快，因此可视图像的失真越小。
- 对比率，对比率的定义是监视器能够产生的最亮色彩（白色）与最暗色彩（黑色）发光度的比率。
- 耗电，以瓦特（W）度量。
- 纵横比，它是横向尺寸与纵向尺寸之比。例如，4∶3是标准纵横比，这样一个宽度为1024像素的显示屏的高度是768像素。宽屏幕显示器的纵横比可以是16∶9，这就意味着一个宽度为1024像素的显示器的高度将是576像素。
- 视角，表示从一个角度来看图像不严重模糊的能力，按照水平和垂直方向来测量。

4. 比较

4.1 CRT

优点：

- 很高的对比度（20,000∶1或者更高，比许多现代LCD和等离子显示器高得多）。
- 高速响应。
- 卓越的色彩，宽色域而低的暗电平。
- 在几乎任何分辨率和刷新速率下都可以提供接近自然的显示。
- 几乎没有色彩、饱和度、对比度或亮度失真，卓越的视角。
- 无输入延迟。
- 可靠的、被证实的显示技术。

缺点：
- 尺寸和质量大（一个 40 英寸的显示器质量超过 200 磅）。
- 在非平面 CRT 中有几何变形。
- 时间长了 CRT 会老化。
- 比同样尺寸的 LCD 更费电。
- 在最高分辨率时显示会有锯齿。
- 不能放置在潮湿环境。
- 如果显像管破裂会有内爆的危险（因为内部真空）。
- 较低刷新率时会明显抖动。

4.2 LCD

优点：
- 非常紧凑而轻便。
- 低能耗。
- 无几何失真。
- 耐用。
- 依靠背光的 LCD 显示器没有或只有一点抖动。

缺点：
- 低对比度。
- 有限的视角。这会引起色彩、饱和度、对比度或亮度改变，即使在预定的视角内仅仅变化一下姿势也会改变。
- 在某些监视器中背光不均匀可能会引起亮度失真，边角尤甚。
- 响应时间短，这可以引起拖尾和重影效果（尽管许多现代 LCD 显示器响应时间已经达到 8 毫秒或更小）。
- 只有一个原始分辨率。显示其他分辨率需要视频转换器，在较低分辨率下显示的图像质量会下降。
- 固定的位深度，许多廉价 LCD 不能显示真彩。
- 输入延迟。
- 比 CRT 稍微贵一点。
- 在生产中可能产生坏点。

4.3 PDP

优点：
- 紧凑而轻便。
- 高对比度（10,000∶1 或更高）。
- 响应速度快。
- 卓越的色彩，宽色域而低的暗电平。
- 几乎没有色彩、饱和度、对比度或亮度失真，卓越的视角。
- 无几何失真。

- 良好的可扩展性，每次扩展面积所需质量较小（从不到 30 英寸宽到世界上最大的 150 英寸）。

缺点：

- 像素间距大，意味着或者分辨率低或者屏幕大。
- 近距离观看时有明显抖动。
- 运行温度高。
- 比 LCD 稍微贵一点。
- 高能耗。
- 只有一个原始分辨率。显示其他分辨率需要视频转换器，在低分辨率下显示的图像质量会下降。
- 固定的位深度。
- 输入延迟。
- 时间长了 PDP 会老化。
- 在生产中可能产生坏点。

Lesson 3

Text

How Operating Systems Work?

If you have a computer, then you have heard about operating systems. Any desktop or laptop PC that you buy normally comes preloaded with Windows XP. Macintosh computers come preloaded with OS (Oprating System) X. Many corporate servers use the Linux or UNIX operating systems. The operating system is the first thing loaded onto the computer — without the operating system, a computer is useless.

More recently, operating systems have started to pop up in smaller computers as well. If you like to tinker with electronic devices, you are probably pleased that operating systems can now be found on many of the devices we use every day, from cell phones to wireless access points. The computers used in these little devices have gotten so powerful that they can now actually run an operating system and applications. The computer in a typical modern cell phone is now more powerful than a desktop computer from 20 years ago, so this progression makes sense and is a natural development. In any device that has an operating system, there's usually a way to make changes to how the device works. One of the reasons operating systems are made out of portable code rather than permanent physical circuits is that they can be changed or modified without having to scrap the whole device.

For a desktop computer user, this means you can add a new security update, system patch, new application or often even a new operating system entirely rather than junk your computer and start again with a new one when you need to make a change. As long as you understand how an operating system works and know how to get at it, you can in many cases change some of the ways it behaves. And, it's as true of your cell phone as it is of your computer.

The purpose of an operating system is to organize and control hardware and software so that the device it lives in behaves in a flexible but predictable way. In this article, we'll tell you what a piece of software must do to be called an operating system, show you how the operating system in your desktop computer works and give you some examples of how to take control of the other operating systems around.

1. What Does It Do?

At the simplest level, an operating system does two things: The first is it manages the hardware and software resources of the system. In a desktop computer, these resources include such things as

the processor, memory, disk space, etc. (On a cell phone, they include the keypad, the screen, the address book, the phone dialer, the battery and the network connection.) The second is it provides a stable, consistent way for applications to deal with the hardware without having to know all the details of the hardware.

The first task, managing the hardware and software resources, is very important, as various programs and input methods compete for the attention of the central processing unit (CPU) and demand memory, storage and input/output (I/O) bandwidth for their own purposes. In this capacity, the operating system plays the role of the good parent, making sure that each application gets the necessary resources while playing nicely with all the other applications, as well as husbanding the limited capacity of the system to the greatest good of all the users and applications.

The second task, providing a consistent application interface, is especially important if there is to be more than one of a particular type of computer using the operating system, or if the hardware making up the computer is ever open to change. A consistent application program interface (API) allows a software developer to write an application on one computer and have a high level of confidence that it will run on another computer of the same type, even if the amount of memory or the quantity of storage is different on the two machines.

Even if a particular computer is unique, an operating system can ensure that applications continue to run when hardware upgrades and updates occur. This is because the operating system and not the application is charged with managing the hardware and the distribution of its resources. One of the challenges facing developers is keeping their operating systems flexible enough to run hardware from the thousands of vendors manufacturing computer equipment. Today's systems can accommodate thousands of different printers, disk drives and special peripherals in any possible combination.

2. What Kinds Are There?

Within the broad family of operating systems, there are generally four types, categorized based on the types of computers they control and the sort of applications they support. The broad categories are:

- Real-time operating system (RTOS) — Real-time operating systems are used to control machinery, scientific instruments and industrial systems. An RTOS typically has very little user interface capability, and no end-user utilities, since the system will be a"sealed box" when delivered for use. A very important part of an RTOS is managing the resources of the computer so that a particular operation executes in precisely the same amount of time every time it occurs. In a complex machine, having a part move more quickly just because system resources are available may be just as catastrophic as having it not move at all because the system is busy.
- Single-user, single task — As the name implies, this operating system is designed to manage the computer so that one user can effectively do one thing at a time. The Palm OS for Palm handheld computers is a good example of a modern single-user, single-task operating system.

- Single-user, multi-task — This is the type of operating system most people use on their desktop and laptop computers today. Microsoft's Windows and Apple's MacOS platforms are both examples of operating systems that will let a single user have several programs in operation at the same time. For example, it's entirely possible for a Windows user to be writing a note in a word processor while downloading a file from the Internet while printing the text of an E-mail message.
- Multi-user — A multi-user operating system allows many different users to take advantage of the computer's resources simultaneously. The operating system must make sure that the requirements of the various users are balanced, and that each of the programs they are using has sufficient and separate resources so that a problem with one user doesn't affect the entire community of users. UNIX, VMS and mainframe operating systems, such as MVS, are examples of multi-user operating systems.

It's important to differentiate here between multi-user operating systems and single-user operating systems that support networking. Windows 2000 and Novell Netware can each support hundreds or thousands of networked users, but the operating systems themselves aren't true multi-user operating systems. The system administrator is the only "user" for Windows 2000 or Netware. The network support and all of the remote user logins the network enables are, in the overall plan of the operating system, a program being run by the administrative user.

With the different types of operating systems in mind, it's time to look at the basic functions provided by an operating system.

3. Wake-Up Call

When you turn on the power to a computer, the first program that runs is usually a set of instructions kept in the computer's read-only memory (ROM). This code examines the system hardware to make sure everything is functioning properly. This power-on self test (POST) checks the CPU, memory, and basic input-output systems (BIOS) for errors and stores the result in a special memory location. Once the POST has successfully completed, the software loaded in ROM (sometimes called the BIOS or firmware) will begin to activate the computer's disk drives. In most modern computers, when the computer activates the hard disk drive, it finds the first piece of the operating system: the bootstrap loader.

The bootstrap loader is a small program that has a single function: It loads the operating system into memory and allows it to begin operation. In the most basic form, the bootstrap loader sets up the small driver programs that interface with and control the various hardware subsystems of the computer. It sets up the divisions of memory that hold the operating system, user information and applications. It establishes the data structures that will hold the myriad signals, flags and semaphores that are used to communicate within and between the subsystems and applications of the computer. Then it turns control of the computer over to the operating system.

The operating system's tasks, in the most general sense, fall into six categories:

- Processor management.

- Memory management.
- Device management.
- Storage management.
- Application interface.
- User interface.

While there are some who argue that an operating system should do more than these six tasks, and some operating-system vendors do build many more utility programs and auxiliary functions into their operating systems, these six tasks define the core of nearly all operating systems.

New Words

laptop	[ˈlæptɔp]	*n.* 便携式电脑，膝上型电脑
normally	[ˈnɔːməli]	*adv.* 正常地，通常地；一般地，经常地
preload	[ˈpriːˈləud]	*v.* 预装
server	[ˈsəːvə]	*n.* 服务器
wireless	[ˈwaiəlis]	*adj.* 无线的
progression	[prəˈgreʃən]	*n.* 进步，进展
modify	[ˈmɔdifai]	*vt.* 更改，修改
scrap	[skræp]	*vt.* 扔弃，拆毁
behave	[biˈheiv]	*v.* 行为
purpose	[ˈpəːpəs]	*n.* 目的，意图，用途
patch	[pætʃ]	*n.* 补丁
		vt. 修补
predictable	[priˈdiktəbl]	*adj.* 可预言的
resource	[riˈsɔːs]	*n.* 资源
keypad	[ˈkiːpæd]	*n.* 数字小键盘
battery	[ˈbætəri]	*n.* 电池
stable	[ˈsteibl]	*adj.* 稳定的
consistent	[kənˈsistənt]	*adj.* 一致的，调和的
attention	[əˈtenʃən]	*n.* 注意
demand	[diˈmɑːnd]	*n.* 要求，需求，需要
		v. 要求，需要
bandwidth	[ˈbændwiθ]	*n.* 带宽
husband	[ˈhʌzbənd]	*vt.* 节俭地使用
confidence	[ˈkɔnfidəns]	*n.* 信心
amount	[əˈmaunt]	*n.* 总数
quantity	[ˈkwɔntiti]	*n.* 量，数量
unique	[juːˈniːk]	*adj.* 唯一的，独特的

occur	[əˈkəː]	*vi.* 发生，出现
challenge	[ˈtʃælindʒ]	*n.* 挑战
		vt. 向……挑战
flexible	[ˈfleksəbl]	*adj.* 柔韧性，灵活的
vendor	[ˈvendɔː]	*n.* 卖主
manufacture	[ˌmænjuˈfæktʃə]	*vt.* 制造，加工
		n. 制造，制造业，产品
accommodate	[əˈkɔmədeit]	*vt.* 适应；供应，供给
peripheral	[pəˈrifərəl]	*adj.* 外围的
		n. 外围设备
combination	[ˌkɔmbiˈneiʃən]	*n.* 结合，联合，合并；组合
execute	[ˈeksikjuːt]	*vt.* 执行，实行
precisely	[priˈsaisli]	*adv.* 正好，恰恰；精确地；准确地
complex	[ˈkɔmpleks]	*adj.* 复杂的，合成的，综合的
available	[əˈveiləbl]	*adj.* 可用到的，可利用的，有用的
catastrophic	[ˌkætəˈstrɔfik]	*adj.* 悲惨的，灾难的
imply	[imˈplai]	*vt.* 暗示，意味
effectively	[iˈfektivli]	*adv.* 有效地，有力地
platform	[ˈplætfɔːm]	*n.* 平台
entirely	[inˈtaiəli]	*adv.* 完全地，彻底地
utility	[juːˈtiliti]	*n.* 实用程序，效用，有用
download	[ˈdaunˌləud]	*v.* 下载
requirement	[riˈkwaiəmənt]	*n.* 需求，要求
balance	[ˈbæləns]	*v.* 平衡
sufficient	[səˈfiʃənt]	*adj.* 充分的，足够的
separate	[ˈsepəreit]	*v.* 分开，隔离
		adj. 分开的，分离的，个别的，单独的
mainframe	[ˈmeinfreim]	*n.* 主机，大型机
differentiate	[ˌdifəˈrenʃieit]	*v.* 区别，区分
network	[ˈnetwəːk]	*n.* 网络
remote	[riˈməut]	*adj.* 遥远的，远程的
enable	[iˈneibl]	*vt.* 使能够
overall	[ˈəuvərɔːl]	*adj.* 全部的，全面的，总的
administrative	[ədˈministrətiv]	*adj.* 管理的，行政的
function	[ˈfʌŋkʃən]	*n.* 功能，作用
		vi. 运行；发挥作用
firmware	[ˈfəːmˌwɛə]	*n.* 固件，韧件（软件硬件相结合）
activate	[ˈæktiveit]	*vt.* 激活；使……活动，对……起作用

subsystem	[ˈsʌbˌsistəm]	*n*. 子系统
division	[diˈviʒən]	*n*. 部分；划分，区分
myriad	[ˈmiriəd]	*n*. 无数，许多
		adj. 无数的，种种的
signal	[ˈsignl]	*n*. 信号
flag	[flæg]	*n*. 标记
		v. 标记
semaphore	[ˈseməfɔː]	*n*. 旗语，信号量
auxiliary	[ɔːgˈziljəri]	*adj*. 辅助的
define	[diˈfain]	*vt*. 定义
core	[kɔː]	*n*. 中心，核心

Phrases

be preloaded with	预先装有
pop up	突然出现
tinker with	胡乱地修补
cell phone	手机，蜂窝电话
wireless access point	无线访问插座
make sense	有意义，很重要
get at	到达，够到，了解
take control of	控制
software developer	软件开发者
address book	地址簿
deal with	安排，处理，涉及
compete for	为……竞争
for one's own purposes	为某人自己的目的
handheld computer	手持电脑
in operation	在运转中；在行动中；在实施中
at the same time	同时
networked user	网络用户
system administrator	系统管理员
network support	网络支持
remote user login	远程用户登录
have in mind	记得，记住
a set of	一套
bootstrap loader	引导装配程序，输入引导子程序
driver program	驱动器程序
data structure	数据结构

utility program　　　　　　　　　　　实用程序

Abbreviations

OS（Operating System）　　　　　　　　操作系统
API（Application Program Interface）　　应用程序接口
RTOS（Real－Time Operating System）　　实时操作系统

Notes

［1］One of the reasons operating systems are made out of portable code rather than permanent physical circuits is that they can be changed or modified without having to scrap the whole device.

本句中，有两个从句，一个从句是 operating systems are made out of portable code rather than permanent physical circuits，作 one of the reasons 的同位语；另一个从句是 that they can be changed or modified without having to scrap the whole device，是一个表语从句，与 is 一起构成谓语。

［2］In this capacity, the operating system plays the role of the good parent, making sure that each application gets the necessary resources while playing nicely with all the other applications, as well as husbanding the limited capacity of the system to the greatest good of all the users and applications.

本句中，making sure that each application gets the necessary resources 为现在分词短语，作结果状语；while playing nicely with all the other applications, as well as husbanding the limited capacity of the system to the greatest good of all the users and applications 作现在分词短语的时间状语，while 表示动作的同时进行，as well as 的意思是“也”，husbanding 与 playing 并列。

［3］A consistent Application Program Interface（API）allows a software developer to write an application on one computer and have a high level of confidence that it will run on another computer of the same type, even if the amount of memory or the quantity of storage is different on the two machines.

本句中，have a high level of confidence 前面省略了 to，even if the amount of memory or the quantity of storage is different on the two machines 是一个让步状语从句，修饰和限定 have a high level of confidence；that it will run on another computer of the same type 是一个同位语从句，对 confidence 作进一步补充说明。

［4］In a complex machine, having a part move more quickly just because system resources are available may be just as catastrophic as having it not move at all because the system is busy.

本句中，having a part move more quickly just because system resources are available 是一个动名词短语，作句子的主语。在该现在分词短语中 just because system resources are available 是一个原因状语从句。as…as…表示比较，意思是“与……一样……”。at all 用于加强语气。原因状语从句 because the system is busy 作动名词短语 having it not move at all 的状语。

Grammar

动词不定式

动词不定式是非谓语动词的一种形式。它在句子中不能作谓语，它没有人称和数的变化。但是，动词不定式又具有动词的许多特点：它可以有自己的宾语、状语及宾语补足语。动词不定式和它的宾语、状语及宾语补足语构成不定式短语。除此之外，动词不定式还有时态和语态的变化。请看下例：

【例】Iron has the property to conduct electric current.

铁具有导电性。

句中，动词不定式短语为 to conduct electric current，electric current 作 conduct 的宾语。

【例】Today we use computers to help us do most of our work.

如今我们使用计算机帮我们做大部分工作。

句中，动词不定式 to help 带有宾语 us 和宾语补足语 do most of our work。

【例】The teacher asked them to work hard.

老师要求他们努力工作。

句中，副词 hard 是不定式 to work 的状语。

动词不定式通常具有名词、形容词、副词的特征，因此可在句中作不同成分。

1. 动词不定式在句中的作用

1）作主语

动词不定式像名词一样作句子的主语。请看下例：

【例】To see once is better than to hear a hundred times.

百闻不如一见。

句中，to see once 是动词不定式短语，在句子中作主语。

【例】To use solar energy is very common nowadays.

如今利用太阳能是很普遍的。

不定式短语作主语时，为了句子的平衡，常常把它放在句尾，而用 it 作形式主语代替不定式放在句首。请看下例：

【例】It's very common to use solar energy nowadays.

如今利用太阳能是很普遍的。

【例】It is necessary to learn C language.

学习 C 语言是很有必要的。

句中，It 是形式主语，而真正的主语是动词不定式 to learn C language。

【例】It took them several years to complete this project.

完成这项工程花了他们几年的时间。

【例】It is recommended to put the computer in a proper place.

建议把计算机放在一个合适的地方。

值得注意的是，当不定式作主语且后面又有一个不定式作表语形成对称结构时，不能使用 it 作形式主语。请看下例：

【例】误：It's to believe to see.

正：To see is to believe.

眼见为实。

若要说明不定式所表示的动作的逻辑主语时，可使用 for 或 of 引导。正确使用介词是关键：当句中的形容词表示人物的性格、品质时，常与介词 of 连用；其他情况用 for 引出。请看下例：

【例】It's wise of them to turn down that proposal.

他们拒绝采纳这个建议是明智的。

【例】It's very kind of you to help us repair this machine.

你帮我们修理机器，真是太好了。

【例】It is very difficult for him to do such an experiment.

他做这样的实验是很困难的。

【例】It is absolutely unnecessary for her to go there.

她完全没有必要到那去。

2）作表语

动词不定式可放在系动词后面作表语。请看下例：

【例】To see is to believe.

眼见为实。

【例】To move something is to do work.

移动物体就是做功。

句中，to do work 是不定式短语作表语；to move something 是不定式短语作主语。

【例】His wish is to become an expert in computer science.

他的志愿是成为一名计算机专家。

【例】What you should do next is to find out where the trouble is.

下一步你应该做的事情是找出故障出在何处。

3）作宾语

动词不定式可作宾语。请看下例：

【例】The new computer which was bought last week will soon begin to work.

上周买的那台新计算机不久将开始运行。

【例】They decided to do the experiment again.

他们决定再次做这个实验。

【例】He tried his best to solve that problem, but failed.

他竭尽全力去解决那个问题，但还是失败了。

当某些动词后面作宾语的不定式必须有自己的补语才能使意思完整时，要用 it 作形式宾语，而将真正的宾语（即不定式）后置。常用这种结构的动词有：think、find、make、

consider、feel 等。请看下例：

【例】They think it necessary to study English.

他们认为必须学习英语。

句中，不定式短语 to study English 作宾语，形容词 necessary 是宾语补足语，it 是形式上的宾语，代替不定式短语 to study English。

【例】A computer makes it easy to work out this kind of problems.

计算机可容易地解决这类问题。

句中，动词不定式短语 to work out this kind of problems 是真正的宾语，it 是形式宾语代替不定式短语。easy 是形容词作宾语补足语。

【例】We consider it our duty to test this software.

我们认为测试这个软件是我们的职责。

句中，our duty 是名词，作宾语补足语。

4）作宾语补足语

某些及物动词要求不定式作宾语补足语。宾语补足语是对宾语的补充说明。请看下例：

【例】I'll get Tom to repair your printer for you.

我去找汤姆帮你修打印机。

句中，动词不定式短语 to repair your printer for you 是宾语 Tom 的补足语，说明“帮你修打印机”的动作是由 Tom 完成的。

【例】A force may cause a body to move.

力可以使物体移动。

句中，a body 是宾语，不定式 to move 是宾语补足语。

【例】We consider electricity to be a form of energy.

我们认为电是一种能量形式。

【例】He considers C language to be very important.

他认为 C 语言十分重要。

当 make、let、have、see、hear、watch、notice、feel 等动词后面用不定式作宾语补足语时，不定式都不带 to。这一点特别重要。请看下例：

【例】I often hear people talk about this kind of printer.

我经常听人们谈论这种打印机。

句中，talk about this kind of printer 是个不带 to 的动词不定式短语，在句中作宾语 people 的补足语。

【例】Please don't forget to have him help you with your computing.

请别忘了让他帮你做运算。

句中，不带 to 的不定式短语 help you with your computing 作 him 的补足语。him 是动词 have 的宾语。

5）作主语补足语

当主动语态的句子变成被动语态时，主动语态句子中的宾语补足语就在被动语态中变成

主语补足语。若主动语态中的宾语补足语由动词不定式构成，则该句变为被动语态后也相应地变为主语补足语。请看下例：

【例】Electricity is considered to be a form of energy by us.

我们认为电是一种能量形式。

句中，动词不定式 to be a form of energy 是对 electricity 的补充说明，在句中作主语补足语。

【例】Light waves are believed to travel faster than sound waves.

人们认为光波比声波传得快。

【例】He was asked to do the experiment at once.

有人请他马上做实验。

但是，当 make、let、have、see、hear、watch、notice、feel 等动词的句子变为被动语态时，原来在主动语态时作宾语补足语的动词不定式这时也变为主语补足语，此时，动词不定式中的 to 不能省略。请看下例：

【例】He was made to finish repairing the printer.

他被迫马上修好打印机。

句中，动词不定式短语 to finish repairing the printer 作主语 he 的补足语。注意，此时 to 不能省略。

【例】He was seen to work over there.

有人看见他在那边工作。

6）作定语

动词不定式作定语时，通常放在它所修饰的名词之后。请看下例：

【例】He never had the chance to learn computer.

他从来没有学习计算机的机会。

句中，to learn computer 是动词不定式，在句中作定语，修饰和限定 the chance。

【例】Scientists are trying to find a way to solve the problem.

科学家们正在试图找到解决该问题的办法。

句中，to solve the problem 作 a way 的定语。

有时，动词不定式与它所修饰的名词是逻辑上的动宾关系。请看下例：

【例】They still have three experiments to do.

他们仍然有三个实验要做。

句中，three experiments 是 do 的逻辑宾语。

【例】We haven't decided which software to buy.

我们还没有决定购买哪个软件。

动词不定式作定语除修饰名词外，还可以修饰代词和数词。请看下例：

【例】Is there anybody to show me the way to the manager's office?

有人给我指一下去经理办公室的路吗？

句中，anybody 是代词。

【例】There is something to do.

还有一些事情要做。

句中，something 是代词，动词不定式 to do 作它的定语。

【例】The professor was the first to arrive yesterday.

昨天教授第一个到达。

句中，first 是数词。

7）作状语

动词不定式可修饰句中的动词、形容词、副词或全句，表示目的、程度、结果、范围、原因等。请看下例：

【例】To install the printer, you must read the manual first.

要安装打印机，必须先读这个手册。（表示目的）

【例】In order to test this theory, we must carry out more experiments.

为了验证这个理论，我们必须做更多的实验。（表示目的）

【例】The temperature is not high enough to change water into steam.

温度不够高，不能使水变成蒸汽。（表示结果或程度）

【例】This hall is big enough to hold two hundred people.

这个礼堂很大，足以容纳 200 人。（表示程度）

【例】We are glad to hear that you've bought a computer.

听说你买了一台计算机，我们十分高兴。（表示原因）

【例】This place is too small to mount so many machines.

这个地方太小，放不下这么多机器。（表示结果）

【例】Some stars are too far to see with our naked eyes.

有些恒星太远，用肉眼看不见。（表示结果）

应该注意，在“too …to…”结构的 to 前面有 not、only、all、but、never 等含有否定意义的词时，后面的不定式就没有否定意义。请看下例：

【例】English is not too difficult to learn.

英语并不难学。

【例】We are only too pleased to accept your kind invitation.

我们很乐意接受你的盛情邀请。

2. 带有连接代词和副词的不定式结构

连接副词 when、where、why、how 和连接代词 what、which、whom 及 whether 等与不定式一起，在句中起名词作用，相当于一个名词从句。它在句子中可以充当以下成分。

1）主语

【例】How to process the data is still a question.

如何处理这些数据，还是一个问题。

【例】Whether to buy the printer or not is not known.

还不知道是否要买打印机。

2）表语

【例】The problem is what to do next?

问题是下一步怎么办?

【例】The problem is where to put the plotter?

问题是把绘图机放在什么地方?

3）宾语

【例】He will tell you which software to buy.

他会告诉你买哪个软件。

【例】Do you know where to put the computer?

你知道把计算机放在什么地方吗?

4）介词宾语

【例】They are at a loss as to how to solve this problem.

他们惊慌失措，不知如何解决这个问题。

【例】We've exchanged views on the problem of which computer to buy.

我们就购买哪台打印机的问题交换了意见。

5）同位语

【例】They have solved the problem what to do next.

他们已经解决了下一步该如何办的问题。

句中，what to do next 作 the problem 的同位语，对 the problem 补充说明。

【例】The problem whether to buy this printer or not has not be decided.

买不买这台打印机，这个问题尚未解决。

句中，whether to buy this printer or not 作 the problem 的同位语。

3. 动词不定式的否定式、完成式、进行式和完成进行式

动词不定式的否定式是在动词不定式的前面加上 not，即加在 to 的前面。请看下例:

【例】The manager told me not to place the computer here.

经理告诉我不要把计算机放在这里。

【例】You'd better not touch the wire with wet hands.

你最好别用湿手摸电线。

句中，you'd 是 you had 的缩写。因为 had better 后面的动词不定式的 to 省略，所以 not 加在 touch 之前。

在英语中，动词词组 had better、would sooner、would rather、need hardly、can't but 之后的动词不定式要省略 to。

动词不定式一般式所表示的动作，通常与谓语所表示的动作同时或在其后发生。请看下例:

【例】I saw him check the printer over there.

我看见他在那边检查打印机。

句中，check 和 saw 同时发生。

【例】I want to learn computer in the future.

我想将来学习计算机。

句中，to learn 发生在 want 之后，表示将要发生的动作。

动词不定式的完成时表示在谓语动词所表示的行为和状态之前所发生的行为，或表示没有实现的行为。它还可以与情态动词一起，表示推测或判断。请看下例：

【例】I seemed to have heard of this professor.

我好像听说过这位教授。

句中，to have heard of 发生在 seemed 之前。

【例】You are lucky to have bought this laser printer.

你运气好，买到了这台激光打印机。

【例】He meant to have repaired the printer.

他原打算修理这台打印机的。

句中表示没有实现的行为，即他没有修理这台打印机。

【例】You should have done the experiment yesterday.

你昨天就该做这个实验。

【例】He may have found out the trouble.

他可能已经找到了故障。

句中，情态动词 + have done 表示推测或判断。

【例】He must have finished the work.

他一定完成了那项工作。

不定式的进行式表示正在进行的动作。请看下例：

【例】The printer seems to be working very well.

这台打印机似乎运行得非常好。

不定式的完成进行式表示在谓语动词所表示的时间之前一直进行的动作。请看下例：

【例】It's a great pleasure to have been working with the professor.

很高兴这段时间和这位教授一起工作。

4. 不定式的被动形式

如果动词不定式的逻辑主语是这个不定式所表示的动作的承受者时，动词不定式一般要用被动形式。有时，逻辑上的主语不出现，但只要在意义上是被动句，不定式仍要用被动式。

【例】Let's show you the software to be tested.

让我们给你看看要测试的软件。

句中，to be tested 是不定式的被动形式。

【例】This form is to be filled in ink.

这张表格要用钢笔填写。

【例】It's a great honour to be invited to attend such a meeting.

能被邀请出席这样的会议是很大的荣誉。

值得注意的是，有时不定式的主动语态具有被动的含义。这时，用不定式的两种语态都

可以，而没有意义上的区别，但主动语态用得较多。请看下例：

【例】There is a lot of work to do.

还有许多工作要做。

【例】Some stars are too far away to see.

一些恒星太远，看不见。

Exercises

一、根据课文内容，判断以下叙述的正误。

(1) Operating systems can only be found on computers.

(2) If you are a desktop computer user, you can add a new security update, system patch, new application or often even a new operating system entirely when you need to make a change.

(3) Managing the hardware and software resources is not as important as providing a consistent application interface.

(4) Today's systems can accommodate thousands of different printers, disk drives but special peripherals in any possible combination.

(5) An RTOS typically has very little user-interface capability, and no end-user utilities.

(6) A good example of a modern single-user, single-task operating system is the Palm OS for Palm handheld computers.

(7) A Windows user can write a note in a word processor while downloading a file from the Internet but can't print the text of an E-mail message at the same time.

(8) UNIX, VMS and mainframe operating systems, such as MVS, are examples of multi-user operating systems.

(9) The operating systems themselves are true multi-user operating systems.

(10) The system administrator is the only "user" for Windows 2000 or Netware.

二、根据课文内容填空。

(1) We should load __________ onto the computer first, and without it a computer is useless.

(2) The purpose of an operating system is ____.

(3) At the simplest level, an operating system does ________________ things: The first is to ____.

(4) One of the challenges facing developers is _________________________ to run hardware from the thousands of vendors manufacturing computer equipment.

(5) In a broad sense, there are generally four types of operating systems. They are: ________________, ________________, ________________ and ______________________.

(6) Real-time operating systems are used to ______________________________________.

(7) ____________________________ is designed to manage the computer so that one user can effectively do one thing at a time.

(8) ____________________ is the type of operating system most people use on their desktop and

laptop computers today.

(9) A multi-user operating system allows ________________ to take advantage of the computer's resources ____________________.

(10) When you turn on the power to a computer, the first program that runs is usually ____________________________.

三、指出下列句子中的动词不定式（短语），并说明其在句子中的作用，然后把它译成汉语。

(1) It took them several years to develop this software.
(2) I really don't know how to test this software.
(3) Later, he found it important to learn C language.
(4) They will have three experiments to do.
(5) Do you know which computer to buy?
(6) AutoCAD is considered to be the best software in CAD.
(7) I heard our manager talk with someone in his office.
(8) How to solve this problem is still a question.
(9) They'll have Mike repair your printer.
(10) In order to find out the virus in the computer, he worked the whole night.
(11) This tool is not advanced enough to solve this problem.
(12) People are trying to find some way to solve this problem.
(13) We are sorry to hear that there is something wrong with your computer.
(14) They told us not to buy this printer.
(15) The problem that we are facing now is how to improve the design.

四、选择与以下各条叙述意义最接近的词汇。

(1) An item in an operation from which the result is obtained.
(2) A graphical form of notation which places each processing step in a "box" and uses arrows to indicate the next step.
(3) A procedure that is activated by another procedure.
(4) A name in a procedure that is used to refer to an argument passed to that procedure.
(5) A sequence of operations carried out repetitively in the same order.

供选择的答案：

A. called routine
B. cycle
C. cycle time
D. flowchart
E. graphs
F. hashing
G. operand
H. parameter

I. protocol

J. program

五、听句子，在画线处填写所听到的单词或词组。

（1） The operating system acts as a host for ________that are run on the machine.

（2） Common contemporary ________include Microsoft Windows, Mac OS, Linux and Solaris.

（3） An operating system is a collection of technologies which are designed to allow the computer to ________certain functions.

（4） Microsoft Windows has a significant majority of market share in the desktop and notebook computer markets, while ________generally run on Linux or other UNIX-like systems.

（5） For hand held and desktop computers, the ________ is generally considered part of the operating system.

（6） Operating systems offer a number of ________to application programs and users.

（7） Almost all computers, including handheld computers, desktop computers, ________, and even video game consoles, use an operating system of some type.

（8） On large ________ like UNIX and UNIX-like systems, the user interface is generally implemented as an application program that runs outside the operating system.

（9） The Microsoft Windows family of operating systems ________as an add-on to the older MS-DOS operating system for the IBM PC.

（10） An operating system is the software component of a computer system that is ________for the management and coordination of activities and the sharing of the resources of the computer.

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

（1） ________: An error can be caused by attempting to divide by 0.

A. Interrupt　B. Default　C. Underflow　D. Overflow

（2） ________: The process of identifying and correcting errors in a program.

A. Debug　B. Bug　C. Fault　D. Default

（3） ________: A collection of related information, organized for easy retrieval.

A. Data　B. Database　C. Buffer　D. Stack

（4） ________: A location where data can be temporarily stored.

A. Area　B. Disk　C. Buffer　D. File

（5） ________: A graphical bar with buttons that perform some of the most common commands.

A. Title bar　B. Tool bar　C. Status bar　D. Scroll bar

（6） Every valid character in a computer that uses even ________must always have an even number of 1 bits.

A. parity　B. check　C. test　D. compare

（7） The maximum number of data that can be expressed by 8 bits is ________.

A. 64　B. 128　C. 255　D. 256

（8） Integration ________is the process of verifying that the components of a system work together

as described in the program design and system design specifications.

A. trying　　B. checking　　C. testing　　D. coding

(9) GIF files are limited to a maximum of 8 bits/pixel, it simply means that no more than 256 colors are allowed in ________.

A. an image　　B. a file　　C. a window　　D. a page

(10) Computer ________is a complex consisting of two or more connected computing units, it is used for the purpose of data communication and resource sharing.

A. storage　　B. device　　C. network　　D. processor

Skill Training

面　　试

面试十分重要。一般来说，雇主会对那些经资料审查基本符合要求的人员发面试通知。但是，接到通知并不意味着一定会被雇用。这取决于面试时受试人表现出来的综合素质（如敬业精神、与人沟通能力、接受力及潜质等）。面试考官可能是一人，也可能是数人，他们通常有相当大的决定权。应按时参加面试，如确有原因不能准时参加，则应事先另约时间。

在李明向 AAA 公司投递应聘简历后的第五天，他收到了 AAA 公司寄来的通知，让他下周二上午去面试。

Tom 是面试考官。假定你是李明，完成下列对话。

Tom: Excuse me, are you Li Ming?

Li: ____.

Tom: Please sit down.

Li: ____.

Tom: I'm rather interested in the report the Personal Department sent me. But there are still some questions I'd like to ask. Have you ever learned database?

Li: ____.

Tom: Are you familiar with FoxPro for Windows?

Li: ____.

Tom: What do you think is the most important in multi-user?

Li: ____.

Tom: How to write a set of database programs?

Li: ____.

Tom: And are you familiar with Network?

Li: ____.

Tom: What kind of Network are you most familiar with?

Li: ____.

Tom: Then, we'd like you to work in our Sales Department for some time first so that you can understand our products and users. Have you any objections?

Li: ____.

Tom: All right. You can come to the Personal Department with this form the day after tomorrow. There are three months for you to work here first, then if both of us are satisfied with each other, we can sign a contract.

Li: ____.

Tom: Good-bye.

Reading Material

Operating System

An operating system is a collection of technologies which are designed to allow the computer to perform certain functions. These technologies may or may not be present in every operating system, and there are often differences in how they are implemented. However, as stated above most modern operating systems are derived from[1] common design ancestors, and are therefore basically similar.

1. Program execution

An operating system's most basic function is to support the running of programs by the users. On a multiprogramming[2] operating system, running programs are commonly referred to as processes. Process management refers to the facilities provided by the operating system to support the creation, execution and destruction[3] of processes, and to facilitate various interactions, and limit others.

The operating system's kernel in conjunction with[4] underlying hardware must support this functionality.

Executing a program involves the creation of a process by the operating system. The kernel creates a process by setting aside or allocating some memory, loading program code from a disk or another part of memory into the newly allocated space, and starting it running.

Operating system kernels store various information about running processes. This information might include:

- A unique identifier, called a process identifier (PID).
- A list of memory the program is using, or is allowed to access.
- The PID of the program which requested its execution, or the parent process ID (PPID).
- The filename and/or path from which the program was loaded.
- A register file, containing the last values of all CPU registers.
- A program counter, indicating the position in the program.

2. Interrupts

Interrupts are central to operating systems as they allow the operating system to deal with the

[1] derive from 得自，由来，衍生
[2] multiprogramming [ˈmʌltiˈprəugræmiŋ] *n.* 多道程序设计
[3] destruction [disˈtrʌkʃən] *n.* 破坏，毁灭
[4] in conjunction with 与……协力

unexpected[1] activities of running programs and the world outside the computer. Interrupt-based programming is one of the most basic forms of time-sharing, being directly supported by most CPUs. Interrupts provide a computer with a way of automatically running specific code in response to events[2]. Even very basic computers support hardware interrupts, and allow the programmer to specify code which may be run when that event takes place.

When an interrupt is received, the computer's hardware automatically suspends[3] whatever program is currently running, and its registers and program counter[4] are saved. This is analogous to placing a bookmark in a book when someone is interrupted by a phone call. This task requires no operating system as such, but only that the interrupt be configured at an earlier time.

In modern operating systems, interrupts are handled by the operating system's kernel. Interrupts may come from either the computer's hardware, or from the running program. When a hardware device triggers an interrupt, the operating system's kernel decides how to deal with this event, generally by running some processing code, or ignoring it. The processing of hardware interrupts is a task that is usually delegated to software called device drivers[5], which may be either part of the operating system's kernel, part of another program, or both. Device drivers may then relay information to a running program by various means.

A program may also trigger an interrupt to the operating system, which are very similar in function. If a program wishes to access hardware for example, it may interrupt the operating system's kernel, which causes control to be passed back to the kernel. The kernel may then process the request which may contain instructions to be passed onto hardware, or to a device driver. When a program wishes to allocate more memory, launch or communicate with another program, or signal that it no longer needs the CPU, it does so through interrupts.

3. Memory Management

Among other things, a multiprogramming operating system kernel must be responsible[6] for managing all system memory which is currently in use by programs. This ensures that a program does not interfere[7] with memory already used by another program. Since programs time share, each program must have independent access to memory.

Cooperative[8] memory management used by many early operating systems assumes that all programs make voluntary[9] use of the kernel's memory manager and do not exceed their allocated memory. This system of memory management is almost never seen anymore since programs often

---

[1] unexpected [ˈʌnikˈspektid] *adj.* 想不到的，意外的，未预料到
[2] event [iˈvent] *n.* 事件
[3] suspend [səsˈpend] *vt.* 挂起，悬挂
[4] counter [ˈkauntə] *n.* 计数器
[5] device driver 设备驱动程序
[6] responsible [risˈpɔnsəbl] *adj.* 可靠的，负责的
[7] interfere [ˌintəˈfiə] *vi.* 干涉，干预，妨碍，打扰
[8] cooperative [kəuˈɔpərətiv] *adj.* 合作的，协力的
[9] voluntary [ˈvɔləntəri] *adj.* 自动的，主动的，自发的

contain bugs which can cause them to exceed their allocated memory. If a program fails[1] it may cause memory used by one or more other programs to be affected or overwritten[2]. Malicious[3] programs or viruses may purposefully[4] alter another program's memory or may affect the operation of the operating system itself. With cooperative memory management it takes only one misbehaved[5] program to crash[6] the system.

Memory protection enables the kernel to limit a process' access to the computer's memory. Various methods of memory protection exist, including memory segmentation[7], and paging[8]. All methods require some level of hardware support which doesn't exist in all computers.

In both segmentation and paging, certain protected mode registers specify to the CPU what memory address it should allow a running program to access. Attempts to access other addresses will trigger an interrupt which will cause the CPU to re-enter supervisor[9] mode, placing the kernel in charge. This is called a segmentation violation or Seg-V for short, and since it is usually a sign of a misbehaving program, the kernel will generally kill the offending program, and report the error.

4. Disk Access and File Systems

Access to files stored on disks is a central feature of all operating systems. Computers store data on disks using files, which are structured in specific ways in order to allow for faster access, higher reliability and to make better use out of the drive's available space. The specific way files are stored on a disk is called a file system, and enables files to have names and attributes[10]. It also allows them to be stored in a hierarchy[11] of directories or folders arranged in a directory tree.

Early operating systems generally supported a single type of disk drive and only one kind of file system. Early file systems were limited in their capacity, speed and in the kinds of file names and directory structures they could use. These limitations often reflected limitations in the operating systems they were designed for, making it very difficult for an operating system to support more than one file system.

While many simpler operating systems support a limited range of options for accessing storage systems, more modern operating systems like UNIX and Linux support a technology known as a virtual file system[12] or VFS. A modern operating system like UNIX supports a wide array of storage devices, regardless of their design or file systems to be accessed through a common application

[1] fail [feil] *vi.* 失败
[2] overwrite [əuvəˈrait] *v.* 覆盖
[3] malicious [məˈliʃəs] *adj.* 怀恶意的，恶毒的
[4] purposefully [ˈpəːpəsfuli] *adv.* 有目的地，自觉地
[5] misbehave [ˈmisbiˈheiv] *v.* （使）行为不端
[6] crash [kræʃ] *n.* &*v.* 崩溃
[7] segmentation [ˌsegmənˈteiʃən] *n.* 分割，分段
[8] paging [ˈpeidʒiŋ] *n.* 内存分页，页面调度
[9] supervisor [ˈsjuːpəvaizə] *n.* 管理员，（网络）超级用户
[10] attribute [əˈtribjuːt] *n.* 属性
[11] hierarchy [ˈhaiərɑːki] *n.* 层次，层级
[12] virtual file system 虚拟文件系统，缩写为 VFS

programming interface (API)[1]. This makes it unnecessary for programs to have any knowledge about the device they are accessing. A VFS allows the operating system to provide programs with access to an unlimited[2] number of devices with an infinite variety of file systems installed on them through the use of specific device drivers and file system drivers.

A connected storage device such as a hard drive will be accessed through a device driver. The device driver understands the specific language of the drive and is able to translate that language into a standard language used by the operating system to access all disk drives. On UNIX this is the language of block devices.

When the kernel has an appropriate device driver in place[3], it can then access the contents of the disk drive in raw format, which may contain one or more file systems. A file system driver is used to translate the commands used to access each specific file system into a standard set of commands that the operating system can use to talk to all file systems. Programs can then deal with these file systems on the basis of filenames and directories/folders contained within a hierarchical structure. They can create, delete, open and close files as well as gathering various information about them, including access permissions[4], size, free space and creation and modification dates.

Various differences between file systems make supporting all file systems difficult. Allowed characters in file names, case sensitivity and the presence of various kinds of file attributes make the implementation of a single interface for every file system a daunting task.

Microsoft Windows presently supports NTFS and FAT[5] file systems, along with network file systems shared from other computers and the ISO 9660 and UDF file systems used for CDs and DVDs. Each file system is usually limited in application to certain media, for example CDs must use ISO 9660 or UDF[6], and as of Windows Vista, NTFS is the only file system which the operating system can be installed on. The NTFS file system is the most efficient and reliable of the Windows file systems, comparing closely in performance to Linux's XFS. Details of its design are not known. Windows Embedded CE 6.0 introduced ExFAT, a file system more suitable for flash drives.

FAT file systems are commonly found on floppy discs, flash memory cards, digital cameras, and many other portable devices because of their relative simplicity. Performance of FAT compares poorly to most other file systems as it uses overly simplistic data structures, making file operations time-consuming. ISO 9660 and Universal Disk Format are two common formats that target Compact Discs and DVDs. Mount Rainier is a newer extension to UDF supported by Linux 2.6 kernels and Windows Vista that facilitates rewriting to DVDs in the same fashion as has been possible with floppy disks.

File systems may provide journaling[7], which provides safe recovery[8] in the event of a

[1] application programming interface (API) 应用程序接口
[2] unlimited [ʌn'limitid] *adj.* 无限制的，无约束的
[3] in place 在适当的位置，适当
[4] permission [pə'miʃən] *n.* 许可，允许
[5] FAT (file allocation table) 文件分配表
[6] UDF (Universal Disk Format) 通用磁盘格式
[7] journaling ['dʒəːnliŋ] *n.* 日志
[8] recovery [ri'kʌvəri] *n.* 恢复

system crash. A journaled file system writes some information twice: first to the journal, which is a log of file system operations, then to its proper place in the ordinary file system. Journaling is handled by the file system driver, and keeps track of each operation taking place that changes the contents of the disk. In the event of a crash, the system can recover to a consistent state by replaying a portion of the journal. Many UNIX file systems provide journaling including ReiserFS, JFS, and Ext3.

In contrast, non-journaled file systems typically need to be examined in their entirety by a utility such as fsck or chkdsk for any inconsistencies[1] after an unclean shutdown. Soft updates is an alternative to journaling that avoids the redundant writes by carefully ordering the update operations. Log-structured file systems and ZFS also differ from traditional journaled file systems in that they avoid inconsistencies by always writing new copies of the data, eschewing[2] in-place updates.

5. Device Drivers

A device driver is a specific type of computer software developed to allow interaction with hardware devices. Typically this constitutes an interface for communicating with the device, through the specific computer bus or communications subsystem that the hardware is connected to, providing commands to and/or receiving data from the device, and on the other end, the requisite interfaces to the operating system and software applications. It is a specialized hardware-dependent computer program which is also operating system specific that enables another program, typically an operating system or applications software package or computer program running under the operating system kernel, to interact transparently with a hardware device and usually provides the requisite interrupt handling necessary for any necessary asynchronous[3] time-dependent hardware interfacing needs.

The key design goal of device drivers is abstraction. Every model of hardware (even within the same class of device) is different. Newer models also are released by manufacturers that provide more reliable or better performance and these newer models are often controlled differently. Computers and their operating systems cannot be expected to know how to control every device, both now and in the future. To solve this problem, OSes essentially dictate[4] how every type of device should be controlled. The function of the device driver is then to translate these OS mandated function calls into device specific calls. In theory a new device, which is controlled in a new manner, should function correctly if a suitable driver is available. This new driver will ensure that the device appears to operate as usual from the operating systems' point of view for any person.

6. Networking

Currently most operating systems support a variety of networking protocols, hardware and

[1] inconsistency [ˌinkənˈsistənsi] *n.* 不一致性，矛盾
[2] eschew [isˈtʃuː] *vt.* 避开，远避
[3] asynchronous [eiˈsiŋkrənəs] *adj.* 不同时的，异步的
[4] dictate [dikˈteit] *v.* 指令，指示，命令，规定

applications for using them. This means that computers running dissimilar[1] operating systems can participate[2] in a common network for sharing resources such as computing, files, printers and scanners using either wired or wireless connections. Networks can essentially allow a computer's operating system to access the resources of a remote computer to support the same functions as it could if those resources were connected directly to the local computer. This includes everything from simple communication, to using networked file systems or even sharing another computer's graphics or sound hardware. Some network services allow the resources of a computer to be accessed transparently, such as SSH which allows networked users direct access to a computer's command line[3] interface.

Client/server networking involves program on a computer somewhere which connects via a network to another computer, called a server. Servers, usually running UNIX or Linux, offer (or host) various services to other network computers and users. These services are usually provided through ports or numbered access points beyond the server's network address. Each port number is usually associated with a maximum of one running program, which is responsible for handling requests to that port. A daemon, being a user program, can in turn access the local hardware resources of that computer by passing requests to the operating system kernel.

参考译文

操作系统是如何工作的?

如果你有台计算机，那么就听说过操作系统了。你买的任何一台台式计算机或笔记本电脑通常都预装了 Windows XP，苹果计算机预装了 OS X。许多服务器都使用 Linux 或 UNIX 操作系统。操作系统（OS）是最先装入计算机的软件———没有操作系统，计算机就不能使用了。

最近，操作系统开始在更小计算机中流行。如果你喜欢摆弄一些电子设备，你或许会高兴地发现我们平常使用的许多设备中都有操作系统，从蜂窝电话到无线访问插座。用在这些小设备中的计算机的功能十分强大，它们现在实际上可以运行操作系统和应用程序。用在最时髦的蜂窝电话中的计算机比 20 年前的台式计算机的功能更强大。所以这个进步非常有意义，也是自然发展的结果。在任何有操作系统的设备中，通常可以通过某种方法来改变该设备的工作方式。操作系统由灵活的代码构成，而不是由固定的物理电路构成的原因之一是它们可以被修改而无须废弃整个设备。

对于台式计算机用户而言，这意味着当你需要改变时，可以增加新的安全更新、系统补丁、新的应用程序或者焕然一新的操作系统，而无须抛弃旧计算机再使用一台新的。只要你理解了操作系统是如何工作的并了解它，在许多情况下你可以改变它的某些行为。计算机是这样的，蜂窝电话也一样。

[1] dissimilar [diˈsimilə] *adj.* 不同的，相异的

[2] participate [pɑːˈtisipeit] *vi.* 参与，参加，分享，分担

[3] command line 命令行

一个操作系统的目的是组织和管理硬件和软件，以便装有它的设备可灵活地、可预见地运行。在本文中，我们将告诉你什么样的软件才能被称之为操作系统，给你展示台式计算机中的操作系统是如何工作的，并举例说明如何管理其他操作系统。

1. 它做什么?

简而言之，操作系统做两件事情。首先管理系统的硬件和软件资源。在台式计算机中，这些资源包括处理器、内存、磁盘空间等（在蜂窝电话上，它们包括小键盘、屏幕、地址簿、电话拨号盘、电池和网络连接器）。其次，它为应用软件操作硬件提供稳定、一致的方法，而无须知道硬件的全部细节。

第一个任务——管理系统的硬件和软件资源——是十分重要的，因为各个程序和输入程序都会为自己的目的来竞争 CPU、所需的内存、存储器及输入/输出带宽。在这种情况下，操作系统扮演了好父母的角色，确保每一个应用都得到它们所需的资源，同时与其他程序运行良好并最大限度地为用户和程序节约系统的有限资源。

当一个操作系统用于多种计算机或计算机的硬件容易更换时，操作系统的第二个任务——提供统一的应用接口——就变得特别重要。统一的应用程序接口（API）使软件开发者在一个计算机上写程序，但很有信心把它运行在同类的计算机上，即便这两个计算机的内存容量不同或存储量不同也没有关系。

即使一个特别的计算机是唯一的，操作系统也能保证当硬件更新和升级时应用程序继续可用。这是因为操作系统而不是应用程序负责管理硬件和资源分配。开发者面对的挑战之一是保持操作系统有足够的灵活性以便运行来自数以千计的计算机设备厂家的硬件。今天的系统可以适应数千种不同的打印机、磁盘驱动器和特殊外部设备。

2. 有哪些种类?

在操作系统这个大家族中，依据它们所管理的计算机类型和它们所支持的应用程序，操作系统通常分为四类。

- 实时操作系统（RTOS）——实时操作系统用来控制机械、科学仪器和工业系统。一个典型的 RTOS 几乎没有什么用户界面性能，也没有终端用户程序，因为交付使用时该系统是一个“密封盒子”。RTOS 最重要的作用是管理计算机的资源，以便每次执行特殊的操作的时间相同。在复杂的计算机系统中，仅仅因为系统资源可用就让某一部分运行较快可能是一个灾难，这和系统忙而不让其运行的结果是一样的。
- 单用户单任务操作系统——如其名所示，这类操作系统用于管理计算机使一个用户能够有效地一次完成一件事情。用于 Palm 手持式计算机的 Palm OS 是一个很好的例子，它是现代的单用户单任务操作系统。
- 单用户多任务操作系统——现在这类操作系统主要使用在台式计算机和笔记本电脑上。微软公司的 Windows 和苹果公司的 MacOS 平台是这类操作系统的两个例子，它让一个用户可以同时运行多个应用程序。例如，一个 Windows 用户完全可以一边用字处理程序写备忘录，一边下载文件，一边打印电子邮件。
- 多用户操作系统——多用户操作系统允许多个用户同时利用计算机资源。该类操作系统必须保证平衡不同用户的需求，并使他们正在使用的每一个程序都有足够的和独立

的资源，以便一个用户的问题不会影响整个用户群。UNIX、VMS 和大型计算机操作系统（如 MVS）都是多用户操作系统的例子。

区分多用户操作系统和支持网络的单用户操作系统很重要。Windows 2000 和 Novell Netware 可以支持成百上千的网络用户，但这些操作系统本身并非真正的多用户操作系统。系统管理员是 Windows 2000 或 Novell Netware 的唯一“用户”。在该操作系统的整体方案中，网络支持和网络可以接受的远程用户登录是一个可以由管理用户使用的程序。

了解了不同类型的操作系统，该看一看操作系统提供的基本功能了。

3. 启动调用

当打开计算机的电源时，运行的第一个程序通常是保存在计算机只读存储器中的一套指令。这个代码检查系统硬件，以便确保各个部件功能正常。开机自检（POST）检查 CPU、内存和基本输入/输出系统是否有误，并把检查结果保存在内存的特别位置。一旦 POST 成功完成，装在 ROM 中的软件（有时也叫作 BIOS 或“固件”）将激活计算机的磁盘驱动器。现在绝大多数的计算机，在计算机激活硬盘驱动器后，它寻找操作系统的第一部分：引导装入程序。

引导装入程序是一个具有单一功能的小程序：它把操作系统装入内存并启动操作系统。在多数基本型号中，引导装入程序建立小的驱动程序，这些驱动程序连接和管理计算机的各种硬件子系统。它建立保存操作系统、用户信息和应用程序的内存分区，而且建立保存大量信号、标志和信号标志的数据结构，用来在计算机的应用程序和子系统内部和之间进行通信，然后把对计算机的控制交给操作系统。

从一般意义说，计算机操作系统的任务可以分为以下六类：

- 处理器管理。
- 内存管理。
- 设备管理。
- 存储管理。
- 应用接口。
- 用户接口。

虽然有人说操作系统的任务应该多于以上六项，也有一些操作系统经销商把许多实用程序和辅助功能加到他们的操作系统中，但这六项任务是所有操作系统的核心。

Lesson 4

Text

Data structure

In computer science, a data structure is a way of storing data in a computer so that it can be used efficiently. Often a carefully chosen data structure will allow the most efficient algorithm to be used. The choice of the data structure often begins from the choice of an abstract data type. A well-designed data structure allows a variety of critical operations to be performed, using as few resources, both execution time and memory space, as possible. Data structures are implemented by a programming language as data types and the references and operations they provide.

Different kinds of data structures are suited to different kinds of applications, and some are highly specialized to certain tasks. For example, B-trees are particularly well-suited for implementation of databases, while networks of machines rely on routing tables to function.

In the design of many types of programs, the choice of data structures is a primary design consideration, as experience in building large systems has shown that the difficulty of implementation and the quality and performance of the final result depends heavily on choosing the best data structure. After the data structures are chosen, the algorithms to be used often become relatively obvious. Sometimes things work in the opposite direction—data structures are chosen because certain key tasks have algorithms that work best with particular data structures. In either case, the choice of appropriate data structures is crucial.

This insight has given rise to many formalized design methods and programming languages in which data structures, rather than algorithms, are the key organizing factor. Most languages feature some sort of module system, allowing data structures to be safely reused in different applications by hiding their verified implementation details behind controlled interfaces. Object-oriented programming languages such as C++ and Java in particular use classes for this purpose.

Since data structures are so crucial, many of them are included in standard libraries of modern programming languages and environments, such as C++'s Standard Template Library containers, the Java Collections Framework, and the Microsoft .NET Framework.

The fundamental building blocks of most data structures are arrays, records, discriminated unions, and references. For example, the nullable reference, a reference which can be null, is a combination of references and discriminated unions, and the simplest linked data structure, the linked list, is built from records and nullable references.

Data structures represent implementations or interfaces: A data structure can be viewed as an

interface between two functions or as an implementation of methods to access storage that is organized according to the associated data type.

1. Array

In computer science an array is a data structure consisting of a group of elements that are accessed by indexing. In most programming languages each element has the same data type and the array occupies a contiguous area of storage. Most programming languages have a built-in array data type.

Some older languages referred to arrays as tables. This was the practice in COBOL.

Some programming languages support array programming (e. g., APL (A Programming Language), newer versions of Fortran) which generalizes operations and functions to work transparently over arrays as they do with scalars, instead of requiring looping over array members.

Multi-dimensional arrays are accessed using more than one index: one for each dimension.

Arrays can be classified as fixed-sized arrays (sometimes known as static arrays) whose size cannot change once their storage has been allocated, or dynamic arrays, which can be resized.

2. Linked list

In computer science, a linked list is one of the fundamental data structures, and can be used to implement other data structures. It consists of a sequence of nodes, each containing arbitrary data fields and one or two references ("links") pointing to the next and/or previous nodes. The principal benefit of a linked list over a conventional array is that the order of the linked items may be different from the order that the data items are stored in memory or on disk, allowing the list of items to be traversed in a different order. A linked list is a self-referential data type because it contains a pointer or link to another datum of the same type. Linked lists permit insertion and removal of nodes at any point in the list in constant time, but do not allow random access. Several different types of linked list exist: singly-linked lists, doubly-linked lists, and circularly-linked lists.

Linked lists can be implemented in most languages. Languages such as Lisp and Scheme have the data structure built in, along with operations to access the linked list. Procedural or object-oriented languages such as C, C++ and Java typically rely on mutable references to create linked lists.

3. Stack

In computer science, a stack is an abstract data type and data structure based on the principle of Last In First Out (LIFO). Stacks are used extensively at every level of a modern computer system. For example, a modern PC uses stacks at the architecture level, which are used in the basic design of an operating system for interrupt handling and operating system function calls. Among other uses, stacks are used to run a Java Virtual Machine, and the Java language itself has a class called "Stack", which can be used by the programmer. The stack is ubiquitous.

A stack-based computer system is one that stores temporary information primarily in stacks,

rather than hardware CPU registers (a register-based computer system).

4. Queue

A queue is a particular kind of collection in which the entities in the collection are kept in order and the principal (or only) operations on the collection are the addition of entities to the rear terminal position and removal of entities from the front terminal position. This makes the queue a First In First Out (FIFO) data structure. In a FIFO data structure, the first element added to the queue will be the first one to be removed. This is equivalent to the requirement that whenever an element is added, all elements that were added before have to be removed before the new element can be invoked. A queue is an example of a linear data structure.

Queues provide services in computer science, transport and operations research where various entities such as data, objects, persons, or events are stored and held to be processed later. In these contexts, the queue performs the function of a buffer.

Queues are common in computer programs, where they are implemented as data structures coupled with access routines, as an abstract data structure or in object-oriented languages as classes. Common implementations are circular buffers and linked lists.

5. Tree

In computer science, a tree is a widely-used data structure that emulates a tree structure with a set of linked nodes. It is an acyclic and connected graph. Most of the literatures also include the constraint that a graph's edges must be undirected to be a tree. In addition to these three constraints, some literature indicates that a graph's edges should be un-weighted to be a tree.

5.1 Nodes

A node may contain a value or a condition or represent a separate data structure or a tree of its own. Each node in a tree has zero or more child nodes, which are below it in the tree (by convention, trees grow down, not up as they do in nature). A node that has a child is called the child's parent node (or ancestor node, or superior node). A node has at most one parent. The height of a node is the length of the longest downward path to a leaf from that node. The height of the root is the height of the tree. The depth of a node is the length of the path to its root (i. e., its root path).

5.1.1 Root nodes

The topmost node in a tree is called the root node. Being the topmost node, the root node will not have parents. It is the node at which operations on the tree commonly begin (although some algorithms begin with the leaf nodes and work up ending at the root). All other nodes can be reached from it by following edges or links. (In the formal definition, each such path is also unique). In diagrams, it is typically drawn at the top. In some trees, such as heaps, the root node has special properties. Every node in a tree can be seen as the root node of the subtree rooted at that node.

5.1.2 Leaf nodes

Nodes at the bottommost level of the tree are called leaf nodes. Since they are at the bottommost level, they do not have any children.

5.1.3 Internal nodes

An internal node or inner node is any node of a tree that has child nodes and is thus not a leaf node.

5.2 Subtrees

A subtree is a portion of a tree data structure that can be viewed as a complete tree in itself. Any node in a tree T, together with all the nodes below it, comprise a subtree of T. The subtree corresponding to the root node is the entire tree; the subtree corresponding to any other node is called a proper subtree (in analogy to the term proper subset).

5.3 Tree ordering

There are two basic types of trees. In an unordered tree, a tree is a tree in a purely structural sense—that is to say, given a node, there is no order for the children of that node. A tree on which an order is imposed—for example, by assigning different natural numbers to each edge leading to a node's children—is called an edge-labeled tree or an ordered tree with data structures built upon them being called ordered tree data structures.

Ordered trees are by far the most common form of tree data structure. Binary trees are one kind of ordered tree.

New Words

carefully	[ˈkɛəfuli]	*adv.* 小心地，谨慎地
algorithm	[ˈælgəriðəm]	*n.* 算法
abstract	[ˈæbstrækt]	*n.* 抽象，摘要，概要
		adj. 抽象的
critical	[ˈkritikəl]	*adj.* 重要的，关键的；苛求的；严重的
operation	[ˌɔpəˈreiʃən]	*n.* 运转，操作
perform	[pəˈfɔːm]	*vt.* 履行，执行
execution	[ˌeksiˈkjuːʃən]	*n.* 实行，完成
implement	[ˈimplimənt]	*n.* 工具，器具
		vt. 执行，贯彻，实现
suit	[sjuːt]	*v.* 合适，适合，相配
particularly	[pəˈtikjuləli]	*adv.* 异常地，显著地
consideration	[kənsidəˈreiʃən]	*n.* 考虑，需要考虑的事项，理由，因素
obvious	[ˈɔbviəs]	*adj.* 明显的，显而易见的

particular	[pəˈtikjulə]	*adj.* 特殊的，特别的，独特的
		n. 细节，详细情况
crucial	[ˈkruːʃəl]	*adj.* 极重要的，有决定性的
insight	[ˈinsait]	*n.* 洞察力，见识
formalize	[ˈfɔːməlais]	*vt.* 使有效，使定形，使形式化
reused	[ˈriːˈjuːzd]	*adj.* 复用的，再生的
hiding	[ˈhaidiŋ]	*n.* 隐匿
environment	[inˈvaiərənmənt]	*n.* 环境，外界
container	[kənˈteinə]	*n.* 容器
framework	[ˈfreimwəːk]	*n.* 构架，框架，结构
fundamental	[ˌfʌndəˈmentl]	*adj.* 基础的，基本的
		n. 基本原则，基本原理
block	[blɔk]	*n.* 块
		vt. 妨碍，阻塞
array	[əˈrei]	*n.* 数组
union	[ˈjuːnjən]	*n.* 联合，联合体，合并
reference	[ˈrefrəns]	*n.* 引用，参考，参照
null	[nʌl]	*adj.* 无效力的，无效的，无价值的，等于零的
		n. 零，空
associate	[əˈsəuʃieit]	*vt.* 联合，关联
index	[ˈindeks]	*n.* 索引，[数学]指数，指标
		vi. 做索引
occupy	[ˈɔkjupai]	*vt.* 占有，占用
contiguous	[kənˈtigjuəs]	*adj.* 邻近的，接近的，毗边的
transparently	[trænsˈpɛərəntli]	*adv.* 显然地，易觉察地
allocate	[ˈæləukeit]	*vt.* 分派，分配
sequence	[ˈsiːkwəns]	*n.* 次序，顺序，序列
arbitrary	[ˈɑːbitrəri]	*adj.* 任意的
pointer	[ˈpɔintə]	*n.* 指针
removal	[riˈmuːvəl]	*n.* 移动，切除
stack	[stæk]	*n.* 堆栈
extensively	[iksˈtensivli]	*adv.* 广阔地
architecture	[ˈɑːkitektʃə]	*n.* 体系，体系结构
interrupt	[intəˈrʌpt]	*v.* 中断
call	[kɔːl]	*v.* 调用
ubiquitous	[juˈbikwitəs]	*adj.* 到处存在的，（同时）普遍存在的
register	[ˈredʒistə]	*n.* 寄存器
procedural	[prəˈsiːdʒərəl]	*adj.* 程序上的，程序性的

mutable	[ˈmjuːtəbl]	*adj.* 可变的，易变的
queue	[kjuː]	*n.* 队列
		vi. 排队
principal	[ˈprinsipəl]	*adj.* 主要的，最重要的
buffer	[ˈbʌfə]	*n.* 缓冲器，缓冲区
routine	[ruˈtiːn]	*n.* 例行程序
acyclic	[eiˈsaiklik]	*adj.* 非循环的，非环状的
undirected	[ˈʌndiˈrektid]	*adj.* 无方向的
node	[nəud]	*n.* 节点
convention	[kənˈvenʃən]	*n.* 约定，习俗，惯例
ancestor	[ˈænsistə]	*n.* 祖先，祖宗
superior	[sjuːˈpiəriə]	*n.* 上级
leaf	[liːf]	*n.* 叶，树叶；页
diagram	[ˈdaiəgræm]	*n.* 图表，图解
subtree	[ˈsʌbtriː]	*n.* 子树
corresponding	[kɔriˈspɔndiŋ]	*adj.* 相应的，对应的
unordered	[ʌnˈɔːdəd]	*adj.* 无序的

Phrases

execution time	运行时间
memory space	内存空间
B-tree	二叉树
data types	数据类型
rely on	依赖，依靠
routing table	路由表
give rise to	引起，使发生
in particular	特别地
Standard Template Library	标准模板库
discriminated union	识别联合，区别联合
linked list	链表
Java Virtual Machine	Java 虚拟机，缩写为 JVM
built in	内置的，建在内部的
along with	连同…… 一起，随同…… 一起
rear terminal position	后端点
front terminal position	前端点
circular buffer	循环缓冲
child node	子节点

parent node	双亲节点，父节点
root node	根节点
leaf node	叶节点
internal node	内节点
ordered tree	有序树
binary tree	二叉树

Abbreviations

APL（A Programming Language）	APL 语言（用于远程终端的一种程序语言，具有处理数组的特别功能）
LIFO（Last In First Out）	后进先出
FIFO（First In First Out）	先进先出

Notes

[1] Most languages feature some sort of module system, allowing data structures to be safely reused in different applications by hiding their verified implementation details behind controlled interfaces.

本句中，feature 作谓语，是一个动词，意思是“具有……的特点”。allowing data structures to be safely reused in different applications by hiding their verified implementation details behind controlled interfaces 是一个现在分词短语，作结果状语。在该状语中，by hiding their verified implementation details behind controlled interfaces 作方式状语。

[2] For example, the nullable reference, a reference which can be null, is a combination of references and discriminated unions, and the simplest linked data structure, the linked list, is built from records and nullable references.

本句中，a reference which can be null 作 the nullable reference 的同位语，对其作补充说明。the linked list 作 the simplest linked data structure 的同位语。

[3] A queue is a particular kind of collection in which the entities in the collection are kept in order and the principal (or only) operations on the collection are the addition of entities to the rear terminal position and removal of entities from the front terminal position.

本句中，in which the entities in the collection are kept in order and the principal (or only) operations on the collection are the addition of entities to the rear terminal position and removal of entities from the front terminal position 是一个定语从句，修饰和限定 a particular kind of collection。keep sth. in order 的意思是“使某事有顺序”，the addition of 的意思是“增加”，removal of 的意思是“去除”。

[4] This is equivalent to the requirement that whenever an element is added, all elements that were added before have to be removed before the new element can be invoked.

本句中，that whenever an element is added, all elements that were added before have to be

removed before the new element can be invoked 是一个同位语从句，说明 the requirement 的具体内容。在该从句中，all elements 是主语；have to be removed 是谓语；whenever an element is added 和 before the new element can be invoked 是两个状语从句，修饰谓语；that were added 是一个定语从句，修饰和限定 all elements。

Grammar

现 在 分 词

分词是非谓语动词的一种，可以分为现在分词和过去分词。现在分词主要起形容词的作用，同时还保留了动词的特征。现在分词也可以有自己的宾语、状语和表语，以构成分词短语。有时它也可有自己单独的逻辑主语。请看下例：

【例】When I entered the room, I found him checking up the printer.

我进屋时，发现他正在检查打印机。

句中，the printer 作 checking up 的宾语。

【例】They saw the manager coming quickly towards them.

他们看见经理正快步向他们走来。

句中，quickly towards them 作 coming 的状语。

【例】Being a manager, he has to decide what to do next.

作为一个经理，他必须决定下一步怎么做。

句中，a manager 作 Being 的表语。

【例】The experiment being over, they began their discussion.

实验做完后，他们开始讨论。

句中，the experiment 作 being over 的逻辑主语。

1. 现在分词的构成

（1）一般情况下，动词后直接加 ing。例如：

work ⟶working

press ⟶pressing

look ⟶looking

enter ⟶entering

（2）对以不发音的 e 结尾的动词，应先去掉 e 再加 ing。例如：

come ⟶coming

invite ⟶inviting

excite ⟶exciting

move ⟶moving

（3）对重读闭音节且动词词尾只有一个辅音字母的动词，该字母双写后再加 ing。例如：

put ⟶putting

get ⟶getting

stop ⟶stopping

top ⟶topping

2. 现在分词在句中的作用

1）作谓语

现在分词作谓语，可用来构成各种进行时态。请看下例：

【例】They are having a meeting now.

他们正在开会。

【例】He was reading a book on computer science this time yesterday.

昨天这个时候，他正在读一本计算机科学方面的书。

【例】They have been working in this famous company all these years.

这些年来，他们一直在这家著名的公司工作。

【例】At last the professor got the result he had been expecting.

最后，教授得到了他盼望的结果。

2）作表语

现在分词作表语时，多表示主语所具有的特征。请看下例：

【例】The theory sounds quite convincing.

这个理论听起来有说服力。

【例】Recent progress in the computer field is quite encouraging.

计算机领域的最新发展，是很鼓舞人心的。

【例】This problem is very pressing.

这个问题很紧迫。

3）作定语

现在分词作定语的时候很多，它通常放在所修饰的名词或代词之前。若是现在分词短语作定语，则通常放在它所修饰的名词或代词之后。请看下例：

【例】This is a moving blackboard.

这是一块移动的黑板。

【例】This is a pressing problem.

这是一个紧迫的问题。

【例】Who is the man standing by the door?

站在门口的那个人是谁?

句中，standing by the door 是现在分词短语，作 the man 的定语。

【例】The professor lived in a room facing the south.

教授住在一间朝南的房间里。

句中，现在分词短语 facing the south 作定语，修饰和限定 a room。

现在分词和现在分词短语作定语时，其意思接近一个定语从句。请看下例：

【例】China is a developing country.

中国是一个发展中国家。

句中，现在分词 developing 修饰 country。a developing country 等于 a country that is a developing。

【例】The man talking with our manager is an expert in computer science.

与我们经理谈话的那个人是一个计算机专家。

句中，现在分词短语 talking with our manager 作定语，修饰 the man。本句可改为：

The man who is talking with our manager is an expert in computer science.

4）作状语

现在分词作状语时，可以表示时间、结果、条件、让步、方式及伴随情况等。请看下例：

【例】Heating water，you can change it into steam.

把水加热，就能把它变成蒸汽。

句中，现在分词短语 heating water 作条件状语，修饰谓语 can change。

现在分词作条件状语时，相当于一个条件状语从句，heating water 等于 if you heat water。

【例】Not knowing what to do，they went to the professor for help.

因为不知道该怎么办，他们找教授帮忙。

句中，现在分词短语 not knowing what to do 作原因状语，修饰谓语 went。

现在分词短语作状语表示原因时，相当于一个原因状语从句，not knowing what to do 等于 as they didn't know what to do。

【例】Being very small，an electron can't be seen by man.

因为电子很小，所以人的肉眼看不见。

句中，being very small 作原因状语，修饰谓语 can't be seen，being very small 等于 as an electron is very small。

【例】Hearing the news that he would be given a rise，he jumped with joy.

听到将要给他加薪的消息，他高兴得跳了起来。

句中，现在分词短语 hearing the news that he would be given a rise 作时间状语。

现在分词短语作时间状语时，相当于 when 引导的时间状语从句。则 hearing the news that he would be given a rise 等于 when he heard the news that he would be given a rise。

【例】The students went into the lab，following the professor.

学生们跟着教授走进了实验室。

句中，following the professor 作方式状语。

【例】They walked into the room，talking and laughing.

他们说着、笑着走进了房间。

句中，talking and laughing 作方式状语。

【例】Her brother wrote a letter to her，saying that he was learning computer.

她弟弟给她写了一封信，说他正在学习计算机。

句中，saying that he was learning computer 作伴随状语。

【例】Although trying his best, he could not finish his work in time.

尽管他竭尽全力，但仍不能按时完成工作。

句中，although trying his best 作让步状语。

【例】Often the system and application programs are linked to these same device drivers, providing continuity in their operation, as well as saving memory space and programming time.

通常这些系统和应用程序都被连接到这些相同的设备驱动程序，这就提供了它们操作的连续性，也节约内存空间和程序时间。

句中，现在分词短语 providing continuity in their operation, as well as saving memory space and programming time 作结果状语。

现在分词短语作状语表示时间时，也可用 when 或 while 加分词短语结构。请看下例：

【例】While working in this company, he learned a lot about computer.

在这家公司工作的时候，他学到了许多计算机方面的知识。

【例】Please do be careful when operating the printer.

操作打印机时一定要小心。

现在分词作状语时，它的逻辑主语一般与句子的主语一致。但有时它也可以有自己独立的逻辑主语，这种结构被称为独立结构。独立结构由名词（或代词）+现在分词（或现在分词短语）构成。它可以表示时间、条件、原因、伴随的动作或情况。请看下例：

【例】The meeting being over, they went home.

会议结束后，他们回家了。

句中，the meeting 作 being over 的逻辑主语。the meeting being over 作时间状语。

【例】This current changing, the magnetic field will change.

如果这一电流发生变化，则磁场也会随之变化。

句中，this current changing 是一独立结构，作条件状语。this current 是 changing 的逻辑主语。

【例】There are two doors, one leading to the living-room, the other leading to the study.

这有两扇门，一扇通往起居室，一扇通往书房。

句中，one leading to the living-room, the other leading to the study 对 two doors 作附加说明。

【例】He is talking to the manager, his eyes sparkling.

他正在和经理谈话，他的目光炯炯有神。

句中，his eyes sparkling 作伴随状语。his eyes 是 sparkling 的逻辑主语。

【例】The manager being away, we have to put the meeting off.

经理不在，我们只好将会议延期。

句中，the manager being away 作原因状语。the manager 是 being away 的逻辑主语。

5）作宾语补足语

有些动词（如 see、find、hear、get、keep、notice 及 watch 等）所要求的宾语补足语可以是现在分词的形式。请看下例：

【例】I saw him running this way.

我看见他正朝这边跑来。

句中，现在分词短语 running this way 作宾语 him 的补足语。

【例】He tried his best to get the printer working again.

他竭尽全力使打印机重新工作。

句中，现在分词短语 working again 作宾语 the printer 的补足语。

【例】I saw her operating the computer.

我看见她正在使用计算机。

【例】The worker kept the machine running at full speed.

这个工人使机器全速运转。

6）作主语补足语

若把现在分词作宾语补足语的句子变为被动语态，则原来的宾语补足语就变成了主语补足语。请看下例：

【例】She was seen operating the computer.

有人看见她正使用计算机。

句中，现在分词短语 operating the computer 这一动作是由主语 she 完成的，故在句中作主语 she 的补足语。

【例】The machine was kept running at full speed by the worker.

这个工人使机器全速运转。

句中，现在分词短语 running at full speed 作主语 the machine 的补足语。

【例】We were kept waiting for a long time that day.

那天，让我们等了很长时间。

3. 现在分词的否定式、完成式和被动式

1）现在分词的否定式

现在分词的否定式是在分词前面加 not。请看下例：

【例】Not knowing how to process data, the young man went to his teacher for help.

因为不知道如何处理数据，这个年轻人就去请教他的老师。

【例】Not having finished his work, he had to stay behind.

由于没有完成工作，他不得不留下来。

2）现在分词的完成式

现在分词的完成式主要用在状语中，它所表示的动作发生在谓语动作之前。请看下例：

【例】Having finished the experiment, they left the laboratory.

做完实验后，他们就离开了实验室。

句中，having finished 发生于 left 之前。

【例】Having learned computer for many years, he knew how to operate a computer quite well.

由于学了多年计算机，他很清楚如何操作计算机。

句中，having learned 发生在 knew 之前。

3）现在分词的被动式

现在分词的被动式具有被动的意思，表示一个动作此时此刻正在进行或与谓语表示的动作同时发生。请看下例：

【例】The building being built is their laboratory.

正在建造的那座楼是他们的实验楼。

句中，being built 是现在分词的被动式，在句中作定语，修饰 the building，表示正在进行。

【例】He knew the man being wanted on the phone.

他认识电话中要寻找的那个人。

句中，knew 与 being wanted 的动作同时发生。

【例】They found the topic being discussed everywhere.

他们发现到处都在讨论这个问题。

Exercises

一、根据课文内容，判断以下叙述的正误。

(1) A data structure is a way of storing data in a computer.

(2) A data structure is suited to all kinds of applications.

(3) A nullable reference is a reference which can be null.

(4) A data structure can only be viewed as an interface between two functions.

(5) Very few programming languages have a built-in array data type.

(6) Linked lists permit insertion and removal of nodes at any point in the list in constant time, but do not allow random access.

(7) Stacks are used at all levels of a modern computer system.

(8) In a FIFO data structure, the first element added to the queue will be the last one to be removed.

(9) The topmost node in a tree is called leaf nodes and will not have children.

(10) Nodes at the bottommost level of the tree are called the root node. They do not have any parents.

二、根据课文内容填空。

(1) B-trees are particularly well-suited for ____________________, while networks of machines rely on ____________________.

(2) The fundamental building blocks of most data structures are ____________, ____________, discriminated unions and ___________.

(3) In computer science an array is a data structure consisting of ______________ that are accessed by ____________________.

(4) Arrays can be classified as ___________ (sometimes known as static arrays) or ___________.

(5) In computer science, a linked list is one of the fundamental _____________, and can be used

to ____________.

(6) A linked list is a self-referential data type because ________________________.

(7) In computer science, a stack is an abstract data type and data structure based on the principle of ________________________.

(8) In computer science, a tree is a widely-used data structure that ________________________ __________.

(9) A node that has a child is called ________________. A node has at most ______________. The height of a node is __________________. The height of the root is ________________. The depth of a node is __________________.

(10) A subtree is a portion of a tree data structure that can be viewed as ____________.

三、指出下列句子中的现在分词（短语），并说明其在句子中的作用，然后把句子译成汉语。

(1) What he said just now is very inspiring.

(2) They came into the room, talking and laughing.

(3) The students doing the experiment are going to leave for Beijing next week.

(4) Being a student, he should work hard to pass the exam.

(5) The discussion being over, they all left the meeting-room.

(6) There being no buses, they had to go back on foot.

(7) When I got home, I found my father repairing my bike.

(8) The space surrounding a magnet is called a magnetic field.

(9) Not having done it right, he tried again.

(10) We have adjusted the moving parts of the machine, thus greatly increasing its speed.

(11) All moving bodies have energy.

(12) He was seen walking out of the manager's office.

(13) This current changing, the magnetic field will change.

(14) Being cooled in the air, the work-piece became hardened.

(15) He told the boy to be careful while crossing the street.

(16) Position the group icon or resize the group window that will receive the new program item so that it occupies one side of your desktop, leaving room for File Manager window.

(17) To do so, make sure the directory containing Excel is included in your AUTOEXEC. BAT path.

(18) Users requiring the full color capabilities of the colour VGA monitor will find that the colour VGA Monitor is a perfect choice.

四、选择与以下各条叙述意义最接近的词汇。

(1) In a computer program, an entity that possesses a value and is known to the program by a name:

(2) An ordered set which contains a fixed number of elements:

(3) To submit a program to a computer for execution:

(4) A secret code used to deny access to unauthorized users:

(5) A large collection of data in support of a data processing tasks:

供选择的答案:

A. data base

B. password

C. keyword

D. array

E. procedure

F. run

G. . data entry

H. variable

I. vector

J. access

K. user name

L. copy

五、听短文，在画线处填写所听到的单词或词组。

A data structure is a specialized ____1____for organizing and storing ____2____. General data structure types include the ____3____, the file, the ____4____, the table, the tree, and so on. Any data structure is designed to organize data to suit a specific purpose so that it can be ____5____ and worked with in appropriate ways. In computer ____6____, a data structure may be selected or designed to store data for the purpose of working on it with various ____7____.

In computer science an array is a data structure consisting of a group of ____8____that are accessed by ____9____. In most programming languages each element has the same data type and the array occupies a contiguous area of ____10____.

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

(1) ________is a device that converts images to digital format.

A. Copier　　B. Printer

C. Scanner　　D. Display

(2) In C language, a ________is a series of characters enclosed in double quotes.

A. matrix　　B. string

C. program　　D. stream

(3) ________ are those programs that help find the information you are trying to locate on the WWW.

A. Windows　　B. Search Engines

C. Web Sites　　D. Web Pages

(4) In C language, ________are used to create variables and are grouped at the top of a program block.

A. declarations B. dimensions

C. comments D. descriptions

(5) An ________ statement can perform a calculation and store the result in a variable so that it can be used later.

A. executable B. input

C. output D. assignment

(6) Each program module is compiled separately and the resulting ________ files are linked together to make an executable application.

A. assembler B. source

C. library D. object

(7) ________ is the conscious effort to make all jobs similar, routine and interchangeable.

A. WWW B. Informatization

C. Computerization D. Standardization

(8) A Web ________ is one of many software applications that function as the interface between a user and the Internet.

A. display B. browser

C. window D. view

(9) Firewall is a ________ mechanism used by organizations to protect their LANs from the Internet.

A. reliable B. stable

C. peaceful D. security

(10) A query is used to search through the database to locate a particular record or records, which conform to specified ________.

A. criteria B. standards

C. methods D. conditions

Skill Training

E-mail 的写作与使用

E-mail 以其方便快捷而广泛使用。在交际中，E-mail 代表个人或组织的形象与风貌，显示水平和实力，直接影响到对方的感受。因此，必须重视 E-mail 的写作。

1. E-mail 的主要内容

E-mail 包括以下五个部分。

1.1 地址

地址包括写信人 E-mail 地址、收信人 E-mail 地址、抄送收信人 E-mail 地址、密送收信人 E-mail 地址，且地址要准确无误。

1.2　标题或主题

主题应体现邮件主旨，要引人注目、意思明确，最好为名词或动名词短语。

1.3　称呼、开头、正文、结尾句

称呼应礼貌得体，符合英语写作习惯。如不知对方姓名只知头衔，可用 Dear + Title 作为称呼；如只知对方姓名不知性别，可用 Dear + 全名；如邮件为一封通函，则用 Dear All 作为邮件称呼。

正文应结构清楚，便于阅读，如正文内容较长，可使用小标题、小段落，或利用星号、下画线及段落间空行等方式使邮件内容清晰。

1.4　礼貌结束语

礼貌结束语是指正文下面的结尾客套话。在非正式的社交信中，常用 Yours 或 Sincerely。假如对方是亲密的朋友，可用 Sincerely yours 等。

1.5　落款

落款包括写信人全名、写信人职务及所属部门、地址、电话号码、传真等。

2. E-mail 写作的五 C 原则

E-mail 写作应该遵循五 C 原则，即 Correctness（准确）、Conciseness（简洁）、Complete（完整）、Clarity（清楚）和 Courtesy（礼貌）原则。

2.1　准确原则

由于 Email 可能涉及双方的权利、义务关系，其准确性至关重要。具体而言，不仅电子邮件的英语语法、标点符号和拼写要做到准确无误，电子邮件内容还要叙述准确，以免引起误会纠纷。

2.2　简洁原则

简洁原则指在不影响完整性和礼貌性的前提下，尽量使用简单句子和简短词语。一封拖沓冗长、措词复杂的电子邮件既浪费写时间，也会给阅读者带来不必要的麻烦。

在具体写作中，英语 E-mail 呈现出的句法特征是句子结构简单明了。词汇特征是用词简单，简明扼要。

为了节约时间和空间，电子邮件中可使用缩略语。如 U 表示 you、Pls 表示 please、Info 表示 information，Qty 表示 quantity 等。

例如：It is very difficult to sell man hole covers in France. This product must have the quality certificate issued in France.

这里两个单句具有明显的因果关系，但书写并不使用表达因果关系的连接词如 since、because。这种舍长句、复合句，选短句、简单句，喜并列弃从属的句法特征在英语 E-mail 写作中十分常见。

例如：Hope you can accept it.

这里使用不完整句，表明与普通商务信函相比，电子邮件更倾向于非正式文体，更为口语化。

2.3　完整原则

电子邮件内容应力求具体、明确、完整，提供读者所需要的信息。邮件是否完整，可以用5W1H来检验，即Who、When、Where、What、Why和How。

2.4　清楚原则

电子邮件应该做到层次清楚，用词准确。

2.5　礼貌原则

写电子邮件时，应该注意措词，语气婉转礼貌，使对方容易接受。这可以通过使用虚拟语气、委婉语气等方法迂回地表达观点，提出要求。

例如：If it is not for the larger orders we receive from a number four regular customers, we could not have quoted for supplies even at that price.

例如：I would appreciate it if you could give me your best quotations for 65,000 pieces.

句中的虚拟语气和委婉语气，既明确了自己的立场，又使对方感到尊重。语气自然诚恳，礼貌得体，易于对方接受。

3. 其他

3.1　附件

如果E-mail带了附件，一定要在正文中提及，告诉对方是什么附件。如果是压缩格式或其他格式，可能还需要告诉对方用什么软件打开。

- 附件文件的名称要有意义，让对方见名识义。
- 如果文件个数过多，应压缩成一个文件包。
- 除非特殊情况，不要使用特别大的附件。
- 特别重要的邮件要加密。

3.2　字体、格式、颜色

E-mail应使用恰当的字体、格式、颜色和文字大小。例如，尽量不要使用红色字体。

3.3　合理使用软件功能

- 如果可能，应打开“拼写检查”，防止单词拼写错误。
- 如果必要，也可以使用“对方收件告知”功能。
- 可以选择立即发送以便快速响应。
- 重要邮件可以在“发件箱”中保存备份。

Reading Material

Sorting[1] Algorithms

One of the fundamental problems of computer science is ordering[2] a list of items. There's a plethora[3] of solutions to this problem, known as sorting algorithms. Some sorting algorithms are simple and intuitive[4], such as the bubble sort. Others, such as the quick sort are extremely complicated, but produce lightening-fast results.

Below are links to algorithms, analysis and source code of the most common sorting algorithms.

1. Sorting Algorithms

The complexity of their algorithms. Algorithmic complexity is a complex subject (imagine that!) that would take too much time to explain here, but suffice it to say that there's a direct correlation[5] between the complexity of an algorithm and its relative efficiency. Algorithmic complexity is generally written in a form known as Big-O notation, where the O represents the complexity of the algorithm and a value n represents the size of the set the algorithm is run against.

For example, $O(n)$ means that an algorithm has a linear[6] complexity. In other words, it takes ten times longer to operate on a set of 100 items than it does on a set of 10 items ($10 \times 10 = 100$). If the complexity was $O(n^2)$ (quadratic[7] complexity), then it would take 100 times longer to operate on a set of 100 items than it does on a set of 10 items.

The two classes of sorting algorithms are $O(n^2)$, which includes the bubble, insertion, selection and shell sorts; and $O(n^{\log n})$ which includes the heap[8], merge[9] and quick sorts.

In addition to algorithmic complexity, the speed of the various sorts can be compared with empirical data. Since the speed of a sort can vary greatly depending on what data set it sorts, accurateempirical results require several runs of the sort be made and the results averaged together. The empirical data on this site is the average of a hundred runs against random[10] a sets on a single-user 250MHz UltraSPARC II. The run times on your system will almost certainly vary from these results, but the relative speeds should be the same—the selection sort runs in roughly[11] half the time

---

[1] sort [sɔːt] *v.* 排序，分类，拣选 *n.* 种类，类别
[2] ordering [ˈɔːdəriŋ] *n.* 排序，分类
[3] plethora [ˈpleθərə] *n.* 过剩，过多
[4] intuitive [inˈtjuːtiv] *adj.* 直觉的，本能的，天生的
[5] correlation [ˌkɔriˈleiʃən] *n.* 相互关系，相关（性）
[6] linear [ˈliniə] *adj.* 线性的
[7] quadratic [kwəˈdrætik] *adj.* 二次的，平方的
[8] heap [hiːp] *n.* 堆 *v.* 堆积
[9] merge [məːdʒ] *v.* 合并
[10] random [ˈrændəm] *adj.* 随机的
[11] roughly [ˈrʌfli] *adv.* 概略地，粗糙地

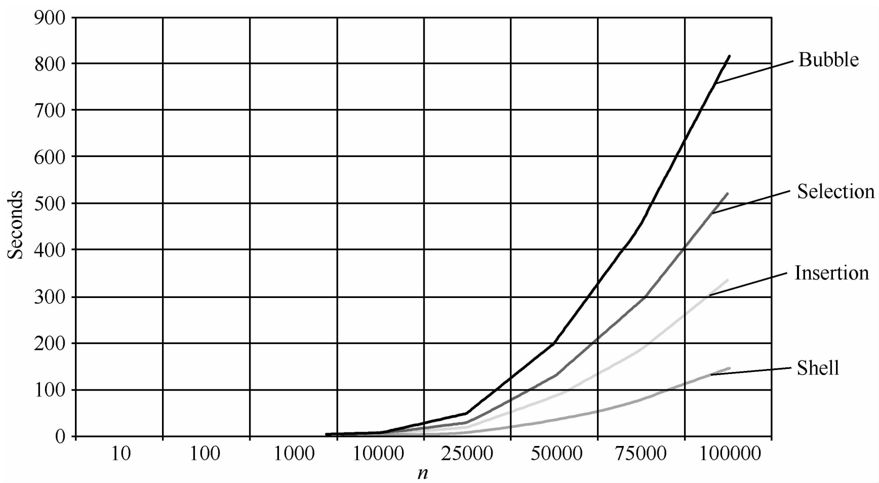
of the bubble sort[1] on the UltraSPARC II, and it should run in roughly half the time on whatever system you use as well.

These empirical efficiency graphs are kind of like golf—the lowest line is the "best". Keep in mind that "best" depends on your situation—the quick sort may look like the fastest sort, but using it to sort a list of 20 items is kind of like going after a fly with a sledgehammer.

1.1 O(n^2) Sorts

As the graph[2] pretty plainly shows, the bubble sort is grossly[3] inefficient, and the shell sort[4] blows it out of the water. Notice that the first horizontal line[5] in the plot area is 100 seconds-these aren't sorts that you want to use for huge amounts of data in an interactive application. Even using the shell sort, users are going to be twiddling their thumbs if you try to sort much more than 10, 000 data items.

On the bright side, all of these algorithms are incredibly[6] simple (with the possible exception[7] of the shell sort). For quick test programs, rapid prototypes, or internal-use software they're not bad choices unless you really think you need split-second efficiency.



1.2 O($n^{\log n}$) Sorts

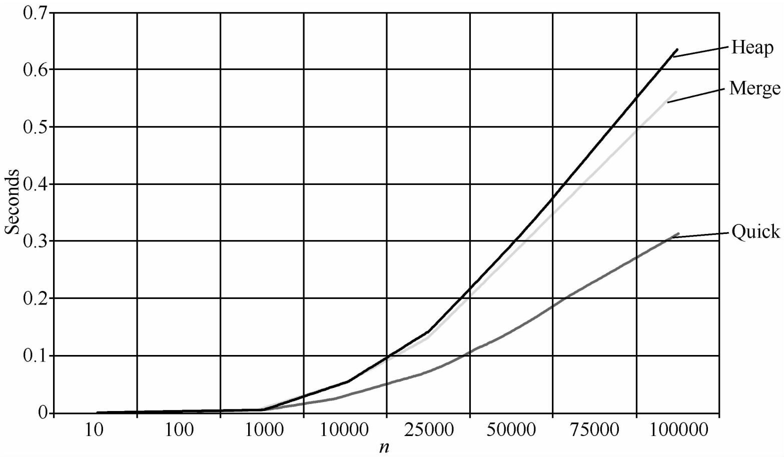
Speaking of split—second efficiency, the O($n^{\log n}$) sorts are where it's at. Notice that the time on this graph is measured in tenths of seconds, instead hundreds of seconds like the O(n^2) graph.

But as with everything else in the real world, there are trade-offs. These algorithms are

---

[1] bubble sort 起泡排序，冒泡排序
[2] graph [grɑːf] *n.* 图表，曲线图
[3] grossly [ˈgrəusli] *adv.* 非常，很
[4] shell sort 希尔排序
[5] horizontal line 水平线
[6] incredibly [inˈkredəbli] *adv.* 难以置信地
[7] exception [ikˈsepʃən] *n.* 例外；除外

blazingly[1] fast, but that speed comes at the cost of complexity. Recursion[2], advanced data s tructures, multiple arrays—these algorithms make extensive use of those nasty things.



In the end, the important thing is to pick the sorting algorithm that you think is appropriate for the task at hand[3]. You should be able to use the source code on this site as a "black box" if you need to—you can just use it, without understanding how it works. Obviously taking the time to understand how the algorithm you choose works is preferable[4], but time constraints are a fact of life.

2. Bubble Sort

2.1 Algorithm Analysis

The bubble sort is the oldest and simplest sort in use. Unfortunately, it's also the slowest.

The bubble sort works by comparing each item in the list with the item next to it, and swapping[5] them if required. The algorithm repeats[6] this process until it makes a pass all the way through the list without swapping any items (in other words, all items are in the correct order). This causes larger values to "bubble" to the end of the list while smaller values "sink[7]" towards the beginning of the list.

The bubble sort is generally considered to be the most inefficient sorting algorithm in common usage. Under best-case conditions (the list is already sorted), the bubble sort can approach a constant O(n) level of complexity. General-case is an abysmal[8] $O(n^2)$.

---

[1] blazingly [ˈbleiziŋli] *adv.* 明显地，显著地，强烈地
[2] recursion [riˈkəːʃən] *n.* 递归，递归式
[3] at hand 在手边，在附近，即将到来
[4] preferable [ˈprefərəbl] *adj.* 更可取的，更好的，更优越的
[5] swapping [ˈswæpiŋ] *n.* 交换
[6] repeat [riˈpiːt] *n.* 重复，反复 *vt.* 重做 *vi.* 重复
[7] sink [siŋk] *vi.* 沉下，(使) 下沉
[8] abysmal [əˈbizməl] *adj.* 非常坏的

While the insertion, selection and shell sorts also have O(n^2) complexities, they are significantly more efficient than the bubble sort.

Pros: Simplicity and ease of implementation.

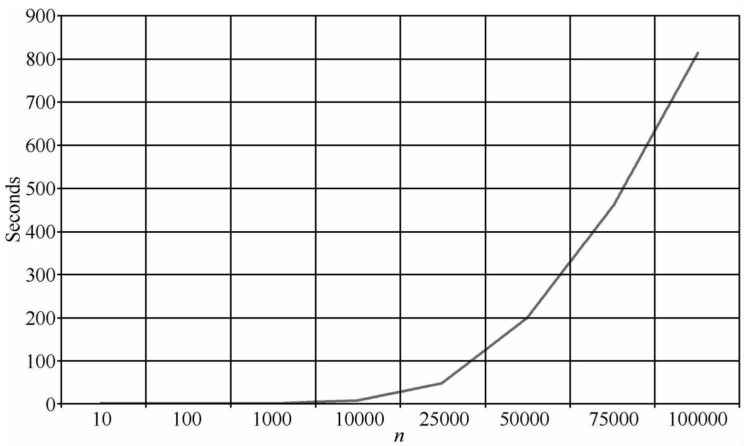
Cons: Horribly[1] inefficient.

2.2 Empirical Analysis

The graph clearly shows the n^2 nature of the bubble sort.

A fair number of algorithm purists[2] (which means they've probably never written software for a living) claim that the bubble sort should never be used for any reason. Realistically, there isn't a noticeable performance difference between the various sorts for 100 items or less, and the simplicity of the bubble sort makes it attractive[3]. The bubble sort shouldn't be used for repetitive sorts or sorts of more than a couple hundred items.

Bubble Sort Efficiency



3. Insertion Sort

3.1 Algorithm Analysis

The insertion sort works just like its name suggests[4]—it inserts each item into its proper place in the final list. The simplest implementation of this requires two list structures—the source list and the list into which sorted items are inserted. To save memory, most implementations use an in-place sort that works by moving the current item past the already sorted items and repeatedly swapping it with the preceding item until it is in place.

Like the bubble sort, the insertion sort has a complexity of O(n^2). Although it has the same complexity, the insertion sort is a little over twice as efficient as the bubble sort.

---

[1] horribly ['hɔrəbli] *adv.* 可怕地，非常地

[2] purist ['pjuərist] *n.* 纯粹主义者

[3] attractive [ə'træktiv] *adj.* 吸引人的，有魅力的

[4] suggest [sə'dʒest] *vt.* 使想起，暗示，建议，提出

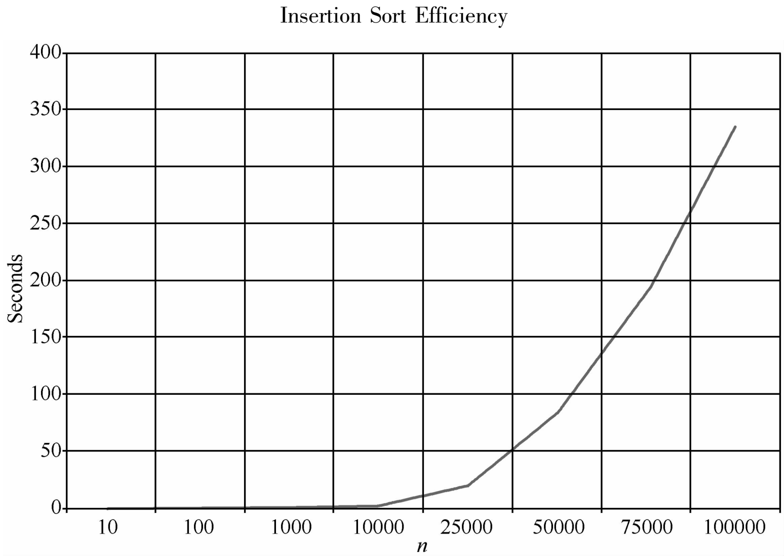
Pros: Relatively simple and easy to implement.

Cons: Inefficient[1] for large lists.

3.2 Empirical Analysis

The graph demonstrates[2] the n^2 complexity of the insertion sort.

The insertion sort is a good middle-of-the-road[3] choice for sorting lists of a few thousand items or less. The algorithm is significantly simpler than the shell sort, with only a small trade-off in efficiency. At the same time, the insertion sort is over twice as fast as the bubble sort and almost 40% faster than the selection sort. The insertion sort shouldn't be used for sorting lists larger than a couple thousand items or repetitive sorting of lists larger than a couple hundred items.



4. Selection Sort

4.1 Algorithm Analysis

The selection sort works by selecting the smallest unsorted[4] item remaining in the list, and then swapping it with the item in the next position to be filled. The selection sort has a complexity of $O(n^2)$.

Pros: Simple and easy to implement.

Cons: Inefficient for large lists, so similar to the more efficient insertion sort that the insertion sort should be used in its place.

---

[1] inefficient [ˌiniˈfiʃənt] *adj.* 效率低的，效率差的

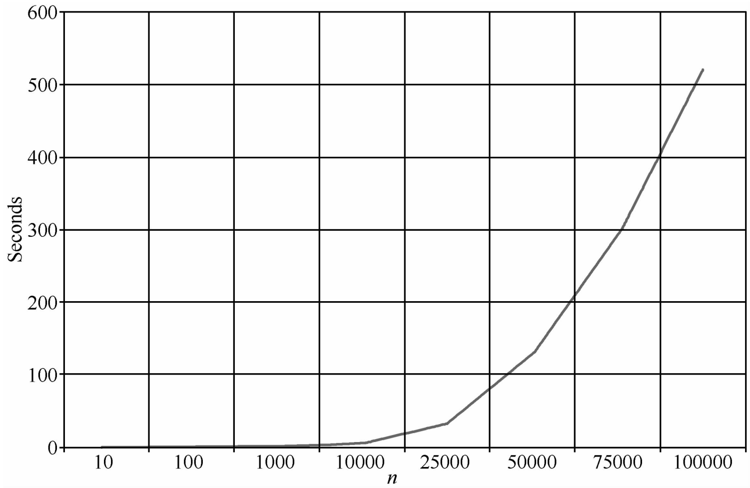
[2] demonstrate [ˈdemənstreit] *v.* 示范，展示，演示

[3] middle-of-the-road 中间道路，中庸的

[4] unsorted [ʌnˈsɔːtid] *adj.* 未排序的

4.2 Empirical Analysis

selection Sort Efficiency



The selection sort is the unwanted stepchild of the n^2 sorts. It yields a 60% performance improvement over the bubble sort, but the insertion sort is over twice as fast as the bubble sort and is just as easy to implement as the selection sort. In short, there really isn't any reason to use the selection sort—use the insertion sort instead.

If you really want to use the selection sort for some reason, try to avoid sorting lists of more than a 1000 items with it or repetitively[1] sorting lists of more than a couple hundred items.

参 考 译 文

数 据 结 构

在计算机科学中，数据结构就是在计算机中存储数据的方法，以便数据可以被有效地使用。通常一个谨慎选择的数据结构允许使用最高效的算法。选择数据结构往往从选择抽象数据类型开始。一个设计优良的数据结构允许执行多种严格的操作，而使用尽可能少的资源(包括执行时间和内存空间)。数据结构由编程语言以数据类型、引用及它们所提供的操作来实现。

不同的数据结构适合不同的应用，并且某些数据结构专门用于特定的任务。例如，二叉树特别适合数据库操作，而计算机网络依赖路由表来运行。

在多种程序设计中，选择数据结构是设计要考虑的一个要素，因为在建立大型系统时的经验已经表明，执行的难点、最终产品的质量和性能与选择良好的数据结构息息相关。选择了数据结构以后，将要使用的算法就相对显而易见了。有时也事与愿违——选择某些数据结构的原因是一些关键任务的算法需要特定的数据结构才能工作得最好。在任意情况下，选择

[1] repetitively [ri'petitivli] *adv.* 重复地

适当的数据结构都至关重要。

这些见解已经促成了许多形式化的设计方法和编程语言，其中数据结构成为主要考虑因素，而不是算法。大多数语言具有一些模块系统的特点，允许通过把已经校验的执行细节隐藏在控制接口后面，使数据结构安全地在不同的应用中重新使用。尤其如 C++ 和 Java 这样的面向对象编程语言，它们使用类来实现这一目标。

因为数据结构如此重要，所以许多数据结构就被包括在现代编程语言和环境的标准库中，如 C++ 的标准模板库、Java 集成框架以及 Microsoft .NET 框架。

大多数数据结构的基本构件是数组、记录、可识别联合和引用。例如，可空引用——一个可以是空的引用——是引用与可识别联合的组合，最简单的链接数据结构——链表——由记录和可空引用来建立。

数据结构表现为实现或接口：可以把数据结构看作是两个函数之间的接口，或者是访问存储数据方法的一种实现，访问存储按照相关数据类型来组织。

1. 数组

在计算机科学中，数组是由一组可以按索引来访问的元素组成的数据结构。在大多数编程语言中，每个元素都有相同的数据类型并且数组占用相邻的存储区域。大部分编程语言都有内置的数组数据类型。

一些比较老的编程语言把数组称为表，在 COBOL 语言中就是如此。

一些编程语言支持数组编程（如 APL、Fortran 的一些新版本）。数组编程概括了透明处理数组的操作和函数，就像处理标量一样，而不再需要对数组元素执行循环。

多维数组用多个索引来访问：每维一个。

数组可以分为一旦分配了存储空间其大小就不能改变的固定大小数组（有时也叫作静态数组）和大小可以改变的动态数组。

2. 链表

在计算机科学中，链表是基本数据结构之一，并可以用来实现其他数据结构。它包括一系列的节点，每个节点都包含专用的数据域以及指向下一个和/或上一个节点的一个或两个引用（“链接”）。与常规数组相比，链表的主要优势是链接元素的顺序可以与存储在内存或磁盘上的数据项顺序不同，从而允许使用不同的顺序遍历元素的列表。链表是自引用数据类型，因为它包含了指向同类其他数据的指针或链接。链表允许在固定时间——在链表的任意一点插入或删除节点，但不允许随机访问。有几种不同类型的链表：单链表、双链表和循环链表。

链表可以使用大多数语言来实现。像 Lisp 和 Scheme 这样的语言有内置的数据结构以及可以访问链表的操作。程序语言或面向对象的语言如 C、C++ 及 Java 通常依赖可变引用来建立链表。

3. 堆栈

在计算机科学中，堆栈是一个抽象数据类型和基于后进先出（LIFO）原则的数据结构。堆栈广泛使用在各级现代计算机系统中。例如，现代 PC 在体系结构级使用堆栈，它被使用

在操作系统的基本设计中，用于中断处理和操作系统功能的调用。堆栈还有其他用途，它可以用来运行 Java 虚拟机，并且 Java 语言本身也有一个称为“堆栈”的类，可以被程序员使用。堆栈无处不在。

基于堆栈的计算机系统主要把临时信息存储在堆栈中，而不是存储在硬件 CPU 的寄存器中（一个基于存储器的计算机系统）。

4. 队列

队列是一种特殊的集合，该集合中的实体有一定的顺序，并且对集合的主要（或唯一）操作是给后端增加实体及从前端删除实体。这就使队列成为先进先出（FIFO）数据结构。在先进先出数据结构中，增加到队列中的第一个元素将被首先删除。这等于要求每增加一个元素，就必须删除以前增加的所有元素才能调用这个新元素。队列是线性数据结构的一个范例。

在计算机科学中，队列提供服务、传输和操作搜索，在队列中可以存储各种实体，如数据、对象、人或者事件以便以后处理。这种情况下，队列的作用就像一个缓冲器。

队列通常用于计算机程序中，在那里它们被当作与访问程序相连的数据结构、抽象数据结构或者面向对象语言中的类来实现。常用的实现是环形缓冲器和链表。

5. 树

在计算机科学中，树是一种广泛使用的数据结构，它用链接的节点集合来模仿一个树形结构。它是一个非环的连接图。许多文献也包括这样的限制：图的边必须是无方向的。还有第三个限制，一些文献指出图的边应该是无权重的。

5.1 节点

一个节点可能包括一个值或一个条件或表现为一个独立的数据结构或者它自身的一个树。树中的每个节点都有零个或多个在其下面的子节点（按照约定，树是朝下生长的，而不是朝上生长的，这与自然界中不一样）。有子节点的节点叫作子节点的双亲节点（或祖先节点或上级节点）。一个节点最多只能有一个双亲节点。一个节点的高度是从该节点向下到叶节点的路径的最大长度。根节点的高度就是树的高度。节点的深度就是它到根节点的路径的长度（如它的根路径）。

5.1.1 根节点

树中最顶上的节点叫作根节点。作为最顶上的节点，根节点没有双亲节点。它是一般树操作开始的节点（虽然有些算法从叶节点开始上溯到根节点结束）。可以从根节点沿下边或链接到达所有其他节点。（按照正式的定义，每个这样的路径都是唯一的）。在图中，它通常画在顶上。在某些树中，例如堆，根节点有特殊功能。树中的每个节点都可以看作以该节点为根的子树的根节点。

5.1.2 叶节点

在树的最底层的节点叫作叶节点。因为在最底层，它们没有子节点。

5.1.3 内节点

一个内节点或内部节点是树中有子节点因而不是叶节点的任何一个节点。

5.2 子树

子树是树数据结构的一部分，它自身可以被看作是一个完整的树。树 T 的任何一个节点下的全部节点组成了树 T 的子树。与根节点相对应的子树是完整树；与任何其他节点相对应的子树是真子树（类似于术语真子集）。

5.3 树的分类

树有两种基本类型。在一个无序树中，树是纯粹结构意义的树，也就是说，有一个节点，该节点的子节点是无序的。一个是顺序强加的树——例如，通过给每个通向节点的子节点的边分配一个自然数——叫作边标记树或有序树，由此建立的数据结构就叫作有序树数据结构。

到目前为止，有序树是最常见的数据结构形式。二叉树就是一种有序树。

Lesson 5

Text

Computer Program

Computer programs (also software programs, or just programs) are instructions for a computer. A computer requires programs to function. Moreover, a computer program does not run unless its instructions are executed by a central processor; however, a program may communicate an algorithm to people without running. Computer programs are usually executable programs or the source code from which executable programs are derived (e.g., compiled).

Computer source code is often written by professional computer programmers. Source code is written in a programming language that usually follows one of two main paradigms: imperative or declarative programming. Source code may be converted into an executable file (sometimes called an executable program or a binary) by a compiler. Alternatively, computer programs may be executed by a central processing unit with the aid of an interpreter, or may be embedded directly into hardware.

Computer programs may be categorized along functional lines: system software and application software. And many computer programs may run simultaneously on a single computer, a process known as multitasking.

1. Programming

Computer programming is the iterative process of writing or editing source code. Editing source code involves testing, analyzing and refining, and sometimes coordinating with other programmers on a jointly developed program. A person who practices this skill is referred to as a computer programmer or software developer. The sometimes lengthy process of computer programming is usually referred to as software development. The term software engineering is becoming popular as the process is seen as an engineering discipline.

1.1 Paradigms

Computer programs can be categorized by the programming language paradigm used to produce them. Two of the main paradigms are imperative and declarative.

Programs written using an imperative language specify an algorithm using declarations, expressions, and statements. A declaration associates a variable name with a datatype. For `example: var x:integer;`. An expression yields a value. For example: 2 + 2 yields 4.

Finally, a statement might assign an expression to a variable or use the value of a variable to alter the program's control flow. For example:

```
x:=2 +2; if x =4 then do_ something ();
```

One criticism of imperative languages is the side-effect of an assignment statement on a class of variables called non-local variables.

Programs written using a declarative language specify the properties that have to be met by the output and do not specify any implementation details. Two broad categories of declarative languages are functional languages and logical languages. The principle behind functional languages is to not allow side-effects, which makes it easier to reason about programs like mathematical functions. The principle behind logical languages (like Prolog) is to define the problem to be solved — the goal — and leave the detailed solution to the Prolog system itself. The goal is defined by providing a list of subgoals. Then each subgoal is defined by further providing a list of its subgoals, etc. If a path of subgoals fails to find a solution, then that path is backtracked and another path is systematically attempted.

The form in which a program is created may be textual or visual. In a visual language program, elements are graphically manipulated rather than textually specified.

1.2 Compilation or interpretation

A computer program in the form of a human-readable, computer programming language is called source code. Source code may be converted into an executable image by a compiler or executed immediately with the aid of an interpreter.

Compiled computer programs are commonly referred to as executables, binary images, or simply as binaries — a reference to the binary file format used to store the executable code. Compilers are used to translate source code from a programming language into either object code or machine code. Object code needs further processing to become machine code, and machine code is the Central Processing Unit's native code, ready for execution.

Interpreted computer programs are either decoded and then immediately executed or are decoded into some efficient intermediate representation for future execution. BASIC (Beginners All-purpose Symbolic Instraction), Perl and Python are examples of immediately executed computer programs. Alternatively, Java computer programs are compiled ahead of time and stored as a machine independent code called bytecode. Bytecode is then executed upon request by an interpreter called a virtual machine.

The main disadvantage of interpreters is computer programs run slower than if compiled. Interpreting code is slower than running the compiled version because the interpreter must decode each statement each time it is loaded and then perform the desired action. On the other hand, software development may be quicker using an interpreter because testing is immediate when the compilation step is omitted. Another disadvantage of interpreters is the interpreter must be present on the computer at the time the computer program is executed. By contrast, compiled computer programs need not have the compiler present at the time of execution.

No properties of a programming language require it to be exclusively compiled or exclusively

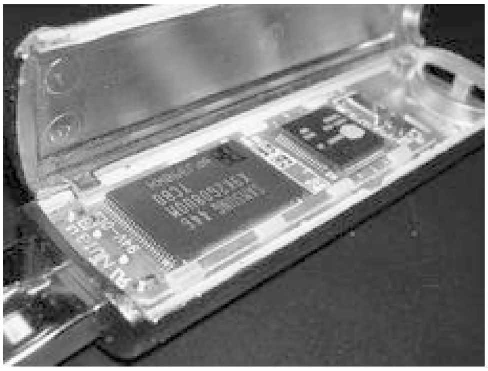
interpreted. The categorization usually reflects the most popular method of language execution. For example, BASIC is thought of as an interpreted language and C a compiled language, despite the existence of BASIC compilers and C interpreters. Some systems use Just-in-time compilation (JIT) whereby sections of the source are compiled "on the fly" and stored for subsequent executions.

2. Execution and storage

Typically, computer programs are stored in non-volatile memory until requested either directly or indirectly to be executed by the computer user. Upon such a request, the program is loaded into random access memory by a computer program called an operating system, where it can be accessed directly by the central processor. The central processor then executes ("runs") the program, instruction by instruction, until termination. A program in execution is called a process. Termination is either by normal self-termination or by error — software or hardware error.

2.1 Embedded programs

Some computer programs are embedded into hardware. A stored-program computer requires an initial computer program stored in its read-only memory to boot. The boot process is to identify and initialize all aspects of the system, from CPU registers to device controllers to memory contents. Following the initialization process, this initial computer program loads the operating system and sets the program counter to begin normal operations. Independent of the host computer, a hardware device might have embedded firmware to control its operation. Firmware is used when the computer program is rarely or never expected to change, or when the program must not be lost when the power is off.



The microcontroller on the right of this USB flash drive is controlled with embedded firmware.

2.2 Manual programming

Computer programs historically were manually input to the central processor via switches. An instruction was represented by a configuration of on/off settings. After setting the configuration, an execute button was pressed. This process was then repeated. Computer programs also historically were manually input via paper tape or punched cards. After the medium was loaded, the starting address was set via switches and the execute button pressed.

Switches for manual input on a Data General Nova 3.

2.3 Automatic program generation

Generative programming is a style of computer programming that creates source code through generic classes, prototypes, templates, aspects and code generators to improve programmer productivity. Source code is generated with programming tools such as a template processor or an Integrated Development Environment. The simplest form of source code generator is a macro processor, such as the C preprocessor, which replaces patterns in source code according to relatively simple rules.

Software engines output source code or markup code that simultaneously becomes the input to another computer process. The analogy is that of one process driving another process, with the computer code being burned as fuel. Application servers are software engines that deliver applications to client computers.

2.4 Simultaneous execution

Many operating systems support multitasking which enables many computer programs to appear to be running simultaneously on a single computer. Operating systems may run multiple programs through process scheduling — a software mechanism to switch the CPU among processes frequently so that users can interact with each program while it is running. Within hardware, modern day multiprocessor computers or computers with multicore processors may run multiple programs.

3. Functional categories

Computer programs may be categorized along functional lines. These functional categories are system software and application software. System software includes the operating system which couples the computer's hardware with the application software. The purpose of the operating system is to provide an environment in which application software executes in a convenient and efficient manner. In addition to the operating system, system software includes utility programs that help manage and tune the computer. If a computer program is not system software then it is application software. Application software includes middleware, which couples the system software with the user interface. Application software also includes utility programs that help users solve application problems, like the need for sorting.

New Words

executable	[ˈeksikjuːtəbl]	*adj.* 可执行的，可实行的
compile	[kəmˈpail]	*vt.* 编译，编辑，汇编
programmer	[ˈprəugræmə]	*n.* 程序设计员
paradigm	[ˈpærədaim]	*n.* 典范，范例，示例，范式

imperative	[imˈperətiv]	*adj.* 命令的，强制的，规则的
declarative	[diˈklærətiv]	*adj.* 宣言的，公布的，陈述的
binary	[ˈbainəri]	*adj.* 二进制的
compiler	[kəmˈpailə]	*n.* 编译器，编译程序
embedded	[emˈbedid]	*adj.* 嵌入的
categorize	[ˈkætigəraiz]	*v.* 分类
simultaneously	[siməlˈteiniəsly]	*adv.* 同时地
multitask	[ˈmʌltiˌtɑːsk]	*n.* 多任务
iterative	[ˈitərətiv]	*adj.* 重复的，反复的，迭代的
refining	[riˈfainiŋ]	*n.* 提炼，改进
coordinating	[kəuˈɔːdineitiŋ]	*vt.* 使协调
lengthy	[ˈleŋθi]	*adj.* 非常长的，冗长的
discipline	[ˈdisiplin]	*n.* 学科；纪律
declaration	[ˌdekləˈreiʃən]	*n.* 声明
expression	[iksˈpreʃən]	*n.* 表达式
statement	[ˈsteitmənt]	*n.* 语句，陈述
variable	[ˈvɛəriəbl]	*n.* 变量
datatype	[ˈdeitətaip]	*n.* 数据类型
yield	[jiːld]	*v.* 产生
assign	[əˈsain]	*v.* 赋值
alter	[ˈɔːitə]	*v.* 改变，变更
principle	[ˈprinsəpl]	*n.* 法则，原则，原理
subgoal	[sʌbgəul]	*n.* 子目的，子目标
backtrack	[ˈbæktræk]	*vi.* 放弃，回溯
systematically	[sistəˈmætikəli]	*adv.* 系统地
textual	[ˈtekstjuəl]	*adj.* 本文的
image	[ˈimidʒ]	*n.* 映像；图像 *vt.* 映射，反映
decode	[ˌdiːˈkəud]	*vt.* 解码，译码
independent	[indiˈpendənt]	*adj.* 独立的
bytecode	[ˈbaitkəud]	*n.* 字节码
omit	[əuˈmit]	*vt.* 省略，疏忽，遗漏
exclusively	[ikˈskluːsivli]	*adv.* 排外地，专有地
process	[prəˈses]	*n.* 进程
prototype	[ˈprəutətaip]	*n.* 原型
macro	[ˈmækrəu]	*n.* 宏
preprocessor	[priːˈprəusesə]	*n.* 预处理程序
engine	[ˈendʒin]	*n.* 发动机，引擎

scheduling	[ˈʃedjuːliŋ]	*n.*	调度
middleware	[ˈmidlwɛə]	*n.*	中间件

Phrases

central processor	中央处理器
source code	源代码
programming language	编程语言
be converted into	被转换为
executable file	可执行文件
central processing unit	中央处理单元
with the aid of	借助于，通过……的帮助
software development	软件开发
software engineering	软件工程
control flow	控制流
assignment statement	赋值语句
non-local variable	非局部变量
functional language	函数语言
visual language	可视化语言
file format	文件格式
object code	目标代码
machine code	机器代码
virtual machine	虚拟机
on the fly	不断地
non-volatile memory	非易失内存
device controller	设备控制器
paper tape	编程纸带
punched card	穿孔卡片
code generator	代码生成器，代码生成程序
Integrated Development Environment	集成开发环境
utility program	实用程序
user interface	用户接口

Abbreviations

BASIC（Beginners All-purpose Symbolic Instruction Code）	BASIC 语言（初学者通用指令码）
JIT（Just-in-time compilation）	即时编译

Notes

[1] Moreover, a computer program does not run unless its instructions are executed by a central processor.

本句中，unless its instructions are executed by a central processor 是一个条件状语从句。unless 的意思是“如果不，除非”，等于 if not。请看下例：

I shall go there tomorrow unless I'm too busy.

如果我不太忙，明天将到那儿去。

[2] Interpreting code is slower than running the compiled version because the interpreter must decode each statement each time it is loaded and then perform the desired action.

本句中，because the interpreter must decode each statement each time it is loaded and then perform the desired action 是一个原因状语从句。在该从句中，each time it is loaded 是一个时间状语从句，修饰谓语 must decode。

[3] Upon such a request, the program is loaded into random access memory by a computer program called an operating system, where it can be accessed directly by the central processor.

本句中，Upon such a request 的意思是“根据这个请求”，called an operating system 是一个过去分词短语作定语，修饰和限定 a computer program。where it can be accessed directly by the central processor 是一个非限定性定语从句，修饰 random access memory，对其作进一步的说明。

[4] Application software includes middleware, which couples the system software with the user interface.

本句中，which couples the system software with the user interface 是一个非限定性定语从句，修饰 middleware，对其作进一步的说明。couple sth. with 的意思是“把……与……连接，把……同……相结合”。

Grammar

过 去 分 词

过去分词和现在分词一样，不能在句中单独作谓语。它主要起形容词和副词的作用，同时还保留了动词的特征。过去分词可以有自己的宾语、状语和补足语，有时也可有自己的逻辑主语。请看下例：

【例】Given enough time, they could finish the work in time.

给他们充足的时间，他们就能按时完成这项工作。

句中，enough time 作 given 的宾语。

【例】Heat used to melt ice is heat of fusion.

用来融化冰的热就是溶解热。

句中，to melt ice 是 used 的状语。

【例】This is the book written by my friend.

这就是我朋友写的那本书。

句中，by my friend 作 written 的状语。

【例】Many things once believed impossible are common today.

许多以前认为不可能的事情如今已经很平常了。

句中，impossible 是补足语。once believed impossible 等于 which were believed impossible。

【例】The manager sat there silently, his head bent low.

经理一声不响低着头坐在那里。

句中，his head 作 bent low 的逻辑主语。

1. 过去分词的构成

1）规则动词

（1）一般情况下，动词后加 ed。例如：

enter ⟶entered

want ⟶wanted

work ⟶worked

（2）对以 e 结尾的动词，直接加 d。例如：

move ⟶moved

use ⟶used

live ⟶lived

（3）重读闭音节且动词词尾只有一个辅音字母时，该辅音字母双写，再加 ed。例如：

plan ⟶planned

stop ⟶stopped

top ⟶topped

（4）以辅音字母加 y 结尾的动词，把 y 变为 i，再加 ed。例如：

try ⟶tried

study ⟶studied

occupy ⟶occupied

2）不规则动词

不规则动词的过去分词需单独记忆。例如：

put ⟶put

break ⟶broken

come ⟶come

do ⟶done

2. 过去分词在句子中的作用

1）作谓语的一部分

过去分词用来构成各种完成时态和被动语态。请看下例：

【例】He has learned C language.

他已经学过 C 语言。

【例】He had learned how to repair printers before he came to work in this company.

在来这家公司之前，他已经学过如何修理打印机了。

【例】The problem was solved at last.

这个问题最后终于解决了。

【例】Two computers will be bought by this school next week.

下周这个学校要买两台计算机。

2）作表语

过去分词作表语时，多表示主语所处的状态。请看下例：

【例】He seemed quite disappointed at the news.

听到这个消息，他似乎很失望。

【例】The door remained locked.

门仍然锁着。

【例】These glasses were broken.

这些玻璃杯是破的。

3）作定语

过去分词和现在分词一样，单独作定语时，通常放在所修饰的名词或代词之前。若是过去短语作定语，则通常放在它所修饰的名词或代词之后。请看下例：

【例】Please take away this broken cup.

请把这个破玻璃杯拿走。

【例】This is a rather complicated problem.

这是一个相当复杂的问题。

【例】Solar energy is a form of energy widely used nowadays.

太阳能是如今广泛使用的一种能量。

句中，widely used nowadays 是过去分词短语，作 energy 的定语。

一个单独的过去分词有时也可以放在它所修饰的名词或代词之后。请看下例：

【例】The experience gained will be of great value to you.

取得的经验对你将很有价值。

句中，gained 修饰 the experience。

过去分词和过去分词短语作定语时，其意义相当于一个定语从句。请看下例：

【例】The professor invited is from Beijing University.

那位被邀请的教授来自北京大学。

句中，invited 作定语，修饰 the professor。本句可改为：

The professor who is invited is from Beijing University.

【例】This is the laboratory built last year.

这是去年建的那个实验室。

句中，过去分词短语 built last year 作定语，修饰 the laboratory。本句可改为：

This is the laboratory which was built last year.

4）作状语

与现在分词一样，过去分词作状语时，可以表示时间、结果、条件、让步、方式及伴随情况等。请看下例：

【例】Thus encouraged, he worked harder than ever.

受到这样的鼓励，他工作得比以往更努力了。

句中，thus encouraged 作原因状语，修饰谓语 worked。

过去分词作状语表示原因时，相当于一个原因状语从句。thus encouraged 等于 as he was thus encouraged。

【例】Seen from space, the earth looks like a water-covered globe.

从太空看，地球像一个被水覆盖的球体。

句中，seen from space 作时间状语，修饰谓语。

过去分词作状语表示时间时，相当于一个时间状语从句。seen from space 等于 when the earth is seen from space.

【例】Given enough time, they could have done it much better.

如果给他们足够的时间，他们会把这件事做得更好。

过去分词作状语表示假设的情况时，相当于一个条件状语从句。given enough time 等于 if they had been enough time。

【例】Compared with the printer he bought last year, this one is much better.

与他去年买的那台打印机相比，这台打印机要好得多。

句中，Compared with the printer he bought last year 作条件状语。

【例】The professor came into the classroom, followed by a group of students.

教授走进教室，身后跟着一群学生。

句中，过去分词短语 followed by a group of students 作方式状语。

过去分词表示“时间”、“条件”时，可在它之前加 when、while、if、unless 等词。请看下例：

【例】When heated to a certain temperature, ice will become water.

当冰加热到一定温度时，就会变成水。

【例】The SCR can be used as a switch, if desired.

需要时，晶闸管可用作开关。

【例】The printer won't print anything, unless operated correctly.

如果操作不正确，打印机就不打印。

过去分词作状语时，它逻辑上的主语一般必须与句子的主语一致。但有时也可以有自己独立的逻辑主语，它也可以表示时间、条件、原因、伴随的动作或情况。请看下例：

【例】The experiment done, they went home.

做完实验之后，他们回家了。

句中，the experiment 作 done 的逻辑主语。the experiment done 作时间状语。

【例】His work finished, he went to have a rest.

工作做完之后，他就去休息了。

句中，his work 作 finished 的逻辑主语，his work finished 在句中作时间状语。

【例】All things considered, your design is better than hers.

考虑到各种因素，你的设计比她的好。

句中，all things 作 considered 的逻辑主语。all things considered 在句中作条件状语。

【例】His legs hurt, he couldn't walk very fast.

他的腿受伤了，走不快。

句中，his legs 作 hurt 的逻辑主语，his legs hurt 在句中作原因状语。

【例】The man walked on, his head held high.

那个男子头抬得高高的，继续向前走。

句中，his head 作 held high 的逻辑主语。his head held high 在句中作方式状语。

5）作宾语补足语

英语中某些动词所要求的宾语补足语可以是过去分词的形式。请看下例：

【例】He is going to have this printer repaired.

他打算去修一下这台打印机。

句中，repaired 作 this printer 的补足语。

应注意，在 have sth. done 词组中，过去分词与宾语之间的逻辑是动宾关系，且过去分词所表示的动作并不是句子的主语发出的。

【例】He couldn't make himself understood.

他不能使别人听懂他说的话。

【例】He found this city greatly changed.

他发现这个城市变化很大。

6）作主语补足语

【例】They should be kept informed of what is going on here.

应该让他们知道这里发生的情况。

句中，过去分词短语 informed of what is going on here 作主语 they 的补足语。

【例】This city was found greatly changed.

人们发现这个城市变化很大。

3. 现在分词与过去分词的区别

（1）现在分词一般有主动的意思，过去分词一般有被动的意思。请看下例：

【例】I heard someone opening the door.

我听见有人开门。

【例】I heard the door opened.

我听见门被打开了。

（2）现在分词作表语时，多表示主语所具有的特征，过去分词则多表示所处的状态。请看下例：

【例】The news is exciting.

这个消息很激动人心。

【例】He was excited at the news.

听到这个消息他很激动。

【例】This cup is broken.

这个茶杯是破的。

（3）现在分词作定语时，一般表示正在进行的动作，表示主动语态；而过去分词作定语时，一般表示已经完成的动作，表示被动语态。请看下例：

【例】He works in a plant manufacturing computer components.

他在一家制造计算机元件的工厂工作。

句中，现在分词短语 manufacturing computer components 作定语，修饰 a plant。本句可改为：

He works in a plant which is manufacturing computer components.

【例】They put the machined parts together.

他们把加工过的零件装配起来。

句中，过去分词 machined 作定语，修饰 parts。本句可改为：

They put the parts which had been machined together.

但是，不少现在分词的一般式和过去分词是没有时间界限的。请看下例：

【例】He lived in a room facing the south.

他住在一间朝南的房间里。

句中，facing the south 是现在分词的一般式，作 room 的定语。本句可改为：

He lived in a room which faced the south.

【例】Computers made in IBM are very favorable.

IBM 公司制造的计算机很受欢迎。

句中，made in IBM 是过去分词短语，作 computers 的定语。本句可改为：

Computers which are made in IBM are very favorable.

（4）现在分词作宾语补足语时，与宾语之间是逻辑上的主谓关系，且表示该动作正在进行；过去分词作宾语补足语时，与宾语之间是逻辑上的动宾关系，有被动的意思。请看下例：

【例】She saw the man operating the computer.

她看见那个人正在操作计算机。

句中，operating the computer 这一动作是由 the man 发出的，所以 the man 是 operating the computer 的逻辑主语。

【例】He is going to have the printer repaired.

他打算把这台打印机送去修理一下。

句中，过去分词 repaired 作宾语 the printer 的补足语，它与 the printer 之间的逻辑关系是动宾关系，即修理打印机。但它们之间也有被动的意思，即打印机被修理。

（5）现在分词被动语态的完成式表示已经进行了的动作，有时也可与过去分词互换。但现在分词被动语态的完成式强调动作的时间概念，而过去分词强调动作的状态。请看下例：

【例】Having been tested many times, the software is found very advanced.

经过多次测试之后，他们发现这个软件非常先进。

句中，现在分词被动语态的完成式 having been tested 表示发生于 is found 之前的动作，

强调动作发生的先后“时间”关系。

【例】Tested many times, the software is found very advanced.

经过多次测试，发现这个软件非常先进。

句中，过去分词 tested 也表示已经进行了的动作，但着重说明该事物的状态。

Exercises

一、根据课文内容，判断以下叙述的正误。

(1) A computer program does not run if its instructions are not executed by a central processor.

(2) Professional computer programmers often write computer source code.

(3) Source code can't be converted into an executable file by a compiler.

(4) Compilers are used to translate object code from a programming language into machine code.

(5) The main disadvantage of interpreters is computer programs run slower than if compiled.

(6) Compiled computer programs need to have the compiler present at the time of execution.

(7) Object code doesn't need further processing to become machine code.

(8) Machine code is the Central Processing Unit's native code, and it is ready for execution.

二、根据课文内容填空。

(1) Source code is written in a programming language that usually follows one of two main paradigms: ____________ or ____________ programming.

(2) The two categories computer programs may be categorized according to their functions are ____________ and ____________.

(3) Editing source code involves ______________, ____________ and refining, and sometimes ____________ on a jointly developed program.

(4) Computer programs can be categorized by the programming language paradigm used to produce them. Two of the main paradigms are ______________________________________.

(5) Programs written using an imperative language specify an algorithm using ______________, ____________ and __________. A declaration associates a variable name with __________.

(6) Programs written using a declarative language do not specify ____________.

(7) Two broad categories of declarative languages are ____________.

(8) A program in execution is called ____________.

(9) Generative programming is a style of computer programming that creates source code through ________________, ________________, ________________, ________________ and ________________ to improve programmer productivity.

(10) According to their functions, computer programs may be categorized as ________________ and ________________.

三、指出下列句子中的过去分词（短语），并说明其在句子中的作用，然后把句子译成汉语。

(1) Several features built into the display provide convenient and comfortable viewing.

(2) The professor came, followed by some of his students.

(3) A small program, called a resident monitor, was created for this purpose.
(4) They were deeply moved by what he said.
(5) John will get this printer repaired.
(6) Given more time, they could have done it better.
(7) Caught in the rain, he was all wet.
(8) The experiment done, they all left.
(9) Homework finished, he ran out of the classroom to play football.
(10) Made of plastics, the machine is light in weight.
(11) Cooled in the air, this kind of steel becomes harder and harder.
(12) That evening, filled with excitement and curiosity, we went to his house.
(13) The long talk was boring and all of us were tired of it.
(14) He was lying on the bed, his hand crossed under his head.
(15) He made an inspiring speech at the meeting held last week.

四、从下列句子中选择5句正确的句子。
(1) The language translator converts the symbolic program into source program.
(2) The coded program written by a programmer is called an object program.
(3) COBOL is self-documenting unlike many technical languages.
(4) FORTRAN has high capabilities for performing input/output operations and in handling nonnumeric data.
(5) Loading is a process through which the information on the diskettes or tapes is read by the input unit and stored in the proper memory location.
(6) The assembler executes source codes directly by determining the meaning of each statement as it is encountered.
(7) Magnetic tape uses cylinders for finding data.
(8) Hexadecimal is used as a shorthand for the internal codes.
(9) Another name of diskette is floppy disk.
(10) An IF-THEN-ELSE selection technique can never be part of a loop.
(11) The most popular language for scientific computing is FORTRAN.
(12) It is much easier to access data in a file than in a database.

五、听短文，在画线处填写所听到的单词或词组。

A programmer, computer programmer, ____1____, coder, or software ____2____, is a person who writes computer software. The term computer programmer can refer to a ____3____ in one area of computer programming or to a generalist who writes code for many kinds of software. One who practices or professes a formal approach to ____4____ may also be known as a programmer analyst. A programmer's primary computer language is often prefixed to the above titles, and those who work in a Web ____5____ often prefix their titles with Web.

Computer programmers write, test, debug, and ____6____ the detailed ____7____, also

called computer programs, which computers must follow to perform their functions. Programmers also conceive, design, and test logical structures for solving problems by computer. Many technical ____8____ in programming — advanced computing technologies and sophisticated new languages and programming tools — have redefined the role of a programmer and ____9____ much of the programming work done today. Job titles and ____10____ may vary, depending on the organization.

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

（1） A sequence of any number of characters enclosed in the double quotes "‘’" is called a character ________.

A. array　　B. group

C. set　　D. string

（2） In C language, when an array name is passed to a function, what is passed is the ________of the beginning of the array.

A. data　　B. value

C. location　　D. element

（3） In C language, the result of the logical ________operator is 1 if the value of its operand is 0, 0 if the value of its operand is non-zero.

A. AND　　B. NOT

C. OR　　D. EOR

（4） A floating constant consists of an integer part, a decimal point, a fraction part, an e or E, and an optionally signed integer ________.

A. exponent　　B. order

C. superfluous　　D. superior

（5） The ________is a temporary storage area that you can use to copy or move selected text or object among applications.

A. cache　　B. pool

C. buffer　　D. clipboard

（6） The ____①____program means a program written in a high-level language. It is generally translated to an ____②____ program, which is in a form directly understandably by the computer. The translation is usually done by a program called ____③____.

① A. assemble　　B. web

C. C　　D. source

② A. object　　B. basic

C. C　　D. assemble

③ A. compiler　　B. assembler

C. compile　　D. transfer

（7） ________processing offers many ways to edit text and establish document formats. You can easily insert, delete, change, move and copy words or blocks of text.

A. Data　　　　　　　　B. Database

C. Word　　　　　　　　D. File

(8) "scrolling" is a technique most commonly associated with ________.

A. disk　　　　　　　　B. display

C. printer　　　　　　　D. memory

Skill Training

公务信函

公务信函属于正式文体，通常用于商务和外贸场合，以及公司与公司之间、公司与客户之间的业务往来等。其中最实用也是最重要的公务信函包括辞职信、请求信、建议信、求职信、道歉信、感谢信、申请信、投诉信、询问信、邀请信和祝贺信。

1. 公务信函的写作格式

当前的格式主要有英式和美式两种。英式结构即缩进式结构，属于比较传统的结构，即每一段的第一行都向里缩进 4 到 8 个字符，且所有段落的缩进距离必须保持一致；称呼顶格写，落款则在中间偏右的位置。美式结构则是齐头式结构，在商务英语中比较流行，尤其普遍运用于电子邮件的交流中，其特点是所有段落的第一行都顶格写，段与段之间空一行，另外称呼、落款等也都顶格写。

2. 公务信函的结构

公务信函一般由以下六部分组成：信头、信内地址、称呼、正文、结尾和签名。

2.1　信头（Heading）

信头是指发信人的地址和日期，通常写在第一页的右上角。行首可以齐头写，也可以逐行缩进写。地址的书写顺序由小到大：门牌号、街道、城市、省（州）、邮编、国名，最后写发信日期。私人信件一般只写寄信日期即可。例如：123 Changan Road Changan District

Xi'an Shanxi Province 727000 P R C Jan. 8, 2013.

2.2　信内地址（Inside Address）

信内地址要写收信人的姓名和地址。该项写在写信日期下一行的左上角，格式与寄信人地址一样。

2.3　称呼（Salutation）

称呼是对收信人的称谓，应与左边线对齐，写在收信人姓名、地址下面 1 ~ 2 行处。在称呼后，英国人常用逗号，美国人则常用冒号。在公务信函中一定要写收信人的姓。大部分信件在称呼前加“Dear”。例如：Dear Professor/Prof. Bergen, Dear Dr. Johnson。

对不相识的人可按性别称呼：Dear Sir：或 Dear Madam。

如果不知收信人的性别，则可用 Dear Sir or Madam。

2.4　正文（Body of Letter）

正文是书信的主体。与中文信件不同的是，英文书信的正文的开头不是先写一些问候语，再阐明写信的目的，而是直接说明写信人的身份及写信的目的，然后提出写信人的情况、想法或要求，并加以必要的解释或说明。英文书信陈述目的时，应该直截了当，意思明确，层次清楚，言简意赅。

书信正文的第一句话或第一段，通常被称为起首语。一般说来，人们习惯用一些客套的写法作为书信正文的起始，即先将对方来信的日期、主题加以简单描述，以便使对方一看便知该信是回答哪一封信的。如果是第一次给别人写信，也可用开头语作必要进行自我介绍，并表明自己写信的主要目的。

2.5　结尾（Complimentary Close）

公务信件的结尾包含两部分：发信人的结尾套语与署名。结尾套语写在签名上面一行，第一个字母要大写，套语结尾后面要加逗号。在公务信函中，发信人常用的结尾套语有：Yours truly，Yours sincerely，Respectfully yours，Cordially yours，Yours cordially 等。

2.6　签名（Signature）

写信人的签名通常位于结尾正下方 1 ~ 2 行。签名一般必须写出全名。签名常常比较潦草，不易辨认，因此在签名的正下方须打印出全名。

应该注意的是，英文的信封写法与中文不同：在信封正面的左上角写发信人姓名、地址。在信封正面中央偏右一点写收信人姓名、地址。

下面是用户写给 AAA 公司的一封信。经理让李明写回信。

假定你是李明，请写出回信。

Export and Import Company
Shanghai
China
October 16, 2001

Director
Dept of Sales
AAA Company
Xi'an
China

Dear Sir:

I bought a computer from your company on the eighth last month. But I find a problem during operating it: When I'm away from the computer for several minutes, I come back to find that the

screen is black and there is nothing on it. But if I press any key on the keyboard, the screen will display the previous information. This has happened several times. I wonder if there's something wrong with the computer. Does it need repairing? I'm looking forward to your reply.

Sincerely yours,
Zhang Qianghua

Reading Material

Web Design

Web design is a process of conceptualization[1], planning, modeling, and execution of electronic media content delivery via Internet in the form of Markup language[2] suitable for interpretation by Web browser and display as Graphical user interface (GUI).

The intent of web design is to create a web site — a collection of electronic files that reside on a web server/servers and present content and interactive features/interfaces to the end user in form of Web pages once requested. Such elements as text, bit-mapped images (GIFs[3], JPEGs[4], PNGs[5]), forms can be placed on the page using HTML[6]/XHTML/XML tags. Displaying more complex media (vector graphics[7], animations, videos, sounds) requires plug-ins such as Flash, QuickTime, Java run-time environment, etc. Plug-ins are also embedded into web page by using HTML/XHTML tags.

Improvements in browsers' compliance[8] with W3C[9] standards prompted a widespread acceptance and usage of XHTML/XML in conjunction with Cascading Style Sheets (CSS[10]) to position and manipulate web page elements and objects. Latest standards and proposals aim at leading to browsers' ability to deliver a wide variety of media and accessibility options to the client possibly without employing plug-ins.

Typically web pages are classified as static or dynamic.

Static pages don't change content and layout[11] with every request unless a human (web master/programmer) manually updates the page.

Dynamic pages adapt their content and/or appearance depending on end-user's input/interaction or

[1] conceptualization [kənˌseptjuəlaiˈzeiʃən] *n.* 概念化
[2] Markup language 标记语言
[3] GIF (Graphics Interchange Format) 图形交换格式
[4] JPEG (Joint Photographic Experts Group) 联合图像专家组
[5] PNG (Portable Network Graphics) 流式网络图形格式
[6] HTML (Hypertext Markup Language) 超文本标识语言
[7] vector graphic 矢量图
[8] compliance [kəmˈplaiəns] *n.* 服从，遵守
[9] W3C (World Wide Web Consortium) 万维网联盟
[10] CSS 层次样式表
[11] layout [ˈleiˌaut] *n.* 布局，设计

changes in the computing environment (user, time, database modifications, etc.) Content can be changed on the client side[1] (end-user's computer) by using client-side scripting languages (JavaScript, JScript, Actionscript, etc.) to alter DOM[2] elements (DHTML[3]). Dynamic content is often compiled on the server utilizing server side scripting languages[4] (Perl, PHP, ASP, JSP, ColdFusion, etc.). Both approaches are usually used in complex applications.

With growing specialization in the information technology field there is a strong tendency to draw a clear line between web design and web development.

1. Web Site Design

A Web site is a collection of information about a particular topic or subject. Designing a website is defined as the arrangement[5] and creation of Web pages that in turn make up a website. A Web page consists of information for which the Web site is developed. A website might be compared to a book, where each page of the book is a web page.

There are many aspects (design concerns) in this process, and due to the rapid development of the Internet, new aspects may emerge[6]. For typical commercial Web sites, the basic aspects of design are:

- The content: The substance, and information on the site should be relevant to the site and should target the area of the public that the website is concerned with.
- The usability: The site should be user-friendly, with the interface and navigation[7] simple and reliable.
- The appearance: The graphics and text should include a single style that flows throughout, to show consistency[8]. The style should be professional, appealing and relevant.
- The visibility: The site must also be easy to find via most, if not all, major search engines[9] and advertisement media.

A Web site typically consists of text and images. The first page of a website is known as the Home page[10] or Index. Some websites use what is commonly called a Splash Page. Splash pages might include a welcome message, language/region selection, or disclaimer. Each web page within a Web site is an HTML file which has its own URL. After each Web page is created, they are typically linked together using a navigation menu[11] composed of hyperlinks. Faster browsing speeds have led to shorter attention spans and more demanding online visitors and this has resulted in less

[1] client side 客户端
[2] DOM (Document Object Module) 文档对象模型
[3] DHTML (Dynamic Hypertext Markup Language) 动态超文本标识语言
[4] scripting language 脚本语言
[5] arrangement [əˈreindʒmənt] *n.* 排列，安排
[6] emerge [iˈməːdʒ] *vi.* 出现，显露
[7] navigation [ˌnæviˈgeiʃən] *n.* 导航
[8] consistency [kənˈsistənsi] *n.* 一致性，连贯性
[9] search engines 搜索引擎
[10] Home page 主页
[11] menu [ˈmenjuː] *n.* 菜单

use of Splash Pages, particularly where commercial websites are concerned.

Once a Web site is completed, it must be published or uploaded[1] in order to be viewable to the public over the Internet. This may be done using an FTP client. Once published, the Web master may use a variety of techniques to increase the traffic, or hits[2], that the website receives. This may include submitting the Web site to a search engine such as Google or Yahoo, exchanging links with other Web sites, creating affiliations with similar Web sites, etc.

2. Multidisciplinary Requirements

Web site design crosses multiple disciplines of information systems, information technology and communication design. The website is an information system whose components are sometimes classified as front-end and back-end. The observable content (e.g., page layout, user interface, graphics, text, audio) is known as the front-end. The back-end comprises the organization and efficiency of the source code[3], invisible scripted functions, and the server-side components that process the output from the front-end. Depending on the size of a Web development project, it may be carried out by a multi-skilled individual (sometimes called a web master), or a project manager may oversee collaborative design between group members with specialized skills.

3. Website Planning

Before creating and uploading a website, it is important to take the time to plan exactly what is needed in the website. Thoroughly considering the audience or target market, as well as defining the purpose and deciding what content will be developed are extremely important.

3.1 Purpose

It is essential to define the purpose of the website as one of the first steps in the planning process. A purpose statement should show focus based on what the website will accomplish and what the users will get from it. A clearly defined purpose will help the rest of the planning process as the audience is identified and the content of the site is developed. Setting short and long term goals for the website will help make the purpose clear and plan for the future when expansion[4], modification, and improvement will take place. Also, goal-setting practices and measurable objectives should be identified to track the progress of the site and determine success.

3.2 Audience

Defining the audience is a key step in the website planning process. The audience is the group of people who are expected to visit your website — the market being targeted. These people will be viewing the website for a specific reason and it is important to know exactly what they are looking for

---

[1] upload [ˈʌpˌləud] v. 上载，上传

[2] hit [hit] n. 点击

[3] source code 源代码

[4] expansion [ikˈspænʃən] n. 扩大，扩张，扩展

when they visit the site. A clearly defined purpose or goal of the site as well as an understanding of what visitors want to do/feel when they come to your site will help to identify[1] the target audience. Upon considering who is most likely to need/use the content, a list of characteristics common to the users such as:

- Audience Characteristics;
- Information Preferences;
- Computer Specifications;
- Web Experience.

Taking into account the characteristics of the audience will allow an effective website to be created that will deliver the desired content to the target audience.

3.3 Content

Content evaluation and organization requires that the purpose of the website be clearly defined. Collecting a list of the necessary content then organizing it according to the audience's needs is a key step in website planning. In the process of gathering the content being offered, any items that do not support the defined purpose or accomplish target audience objectives should be removed. It is a good idea to test the content and purpose on a focus group and compare the offerings to the audience needs. The next step is to organize the basic information structure by categorizing the content and organizing it according to user needs. Each category should be named with a concise[2] and descriptive[3] title that will become a link on the website. Planning for the site's content ensures that the wants/needs of the target audience and the purpose of the site will be fulfilled.

3.4 Compatibility and restrictions

Because of the market share of modern browsers (depending on your target market), the compatibility of your website with the viewers is restricted[4]. For instance, a website that is designed for the majority of websurfers will be limited to the use of valid XHTML 1.0 Strict or older, Cascading Style Sheets Level 1, and 1024 × 768 display resolution. This is because Internet Explorer is not fully W3C standards compliant with the modularity of XHTML[5] 1.1 and the majority of CSS beyond 1. A target market of more alternative browser (e.g., Firefox and Opera) users allow for more W3C compliance and thus a greater range of options for a web designer.

Another restriction on webpage design is the use of different Image file formats. The majority of users can support GIF, JPEG, and PNG (with restrictions). Again Internet Explorer is the major restriction here, not fully supporting PNG's advanced transparency features, resulting in the GIF format still being the most widely used graphic file format for transparent[6] images.

[1] identify [aiˈdentifai] *v.* 识别，鉴别，确定
[2] concise [kənˈsais] *adj.* 简明的，简练的
[3] descriptive [disˈkriptiv] *adj.* 描述的，叙述的
[4] restricted [risˈtriktid] *adj.* 受限制的
[5] XHTML (eXtensible HTML) 可扩展超文本标识语言
[6] transparent [trænsˈpɛərənt] *adj.* 透明的

Many website incompatibilities go unnoticed by the designer and unreported by the users. The only way to be certain a website will work on a particular platform is to test it on that platform.

3.5 Planning documentation

Documentation is used to visually plan the site while taking into account the purpose, audience and content, to design the site structure, content and interactions that are most suitable for the website. Documentation may be considered a prototype for the website — a model which allows the website layout to be reviewed, resulting in suggested changes, improvements and/or enhancements. This review process increases the likelihood[1] of success of the website.

First, the content is categorized and the information structure is formulated. The information structure is used to develop a document or visual diagram called a site map[2]. This creates a visual of how the web pages will be interconnected, which helps in deciding what content will be placed on what pages. There are three main ways of diagramming[3] the website structure:

- Linear Website Diagrams will allow the users to move in a predetermined sequence;
- Hierarchical structures (of Tree Design Website Diagrams) provide more than one path for users to take to their destination;
- Branch Design Website Diagrams allow for many interconnections between web pages such as hyperlinks within sentences.

In addition to planning the structure, the layout and interface of individual pages may be planned using a storyboard. In the process of storyboarding, a record is made of the description, purpose and title of each page in the site, and they are linked together according to the most effective and logical diagram type. Depending on the number of pages required for the website, documentation methods may include using pieces of paper and drawing lines to connect them, or creating the storyboard using computer software.

Some or all of the individual pages may be designed in greater detail as a website wireframe[4], a mock up model or comprehensive layout of what the page will actually look like. This is often done in a graphic program, or layout design program. The wireframe has no working functionality, only planning.

参 考 译 文

计算机程序

计算机程序（也叫作软件程序，或者就叫作程序）是计算机的指令。计算机需要程序来工作。而且，除非计算机程序的指令被中央处理器执行，否则它就不能运行。然而，程序

[1] likelihood [ˈlaiklihud] *n.* 可能，可能性
[2] site map 网站地图
[3] diagramming [ˈdaiəˌɡræmiŋ] *n.* 图样，图表
[4] website wireframe 网站线框架

不运行也能给人传达算法。计算机程序通常是可执行程序或能够产生（如经过编译）可执行程序的源代码。

计算机源代码通常由专业计算机程序员编写。计算机源代码使用一种编程语言来编写，编程语言通常可以是以下两种主要范式之一：命令式编程或说明式编程。源代码也可以被编译器转换为可执行文件（有时也叫作可执行程序或二进制文件）。另外，计算机程序可以借助解释器被中央处理器执行，或者直接嵌入到硬件中。

计算机程序可以按照功能分为以下种类：系统软件和应用软件。多个计算机程序可以同时运行在一个计算机中—— 一个被称为多任务处理的过程。

1. 编程

计算机编程是编写或编辑源代码的反复过程。编辑源代码包括测试、分析和改进，并且有时也要与联合开发程序的其他程序员协作。进行这样技能活动的人就叫作计算机程序员或软件开发者。有时漫长的计算机编程过程通常称为软件开发。因为这个过程被看作是一个工程学科，所以术语软件工程开始流行。

1.1 范式

计算机程序可以按照用来编写它们的编程语言范式来分类。两个主要范式是命令式编程和说明式编程。

用命令式语言编写的程序用声明、表达式和语句来详细说明算法。一个声明把一个变量名与数据类型相联系。例如：`var x：integer;`。一个表达式产生一个值。例如：2 +2 等于 4。

最后，一个语句可以把一个表达式的值赋给一个变量或者用这个变量的值改变程序的控制流程。例如：

```
x： =2 +2； if x = 4 then do_ something ();
```

对命令式语言的一条批判性意见是一个赋值语句对一类叫作非本地变量的变量的副效应。

用说明式语言编写的程序详细说明输出必须满足的特性而不说明任何实现的细节。说明式语言的两个主要种类是函数语言和逻辑语言。函数语言背后的原则是不允许副效应，这使得程序更像数学函数。逻辑语言（如 Prolog）背后的原则是定义要解决的问题——目标——把详细的解决方案留给 Prolog 系统自己。通过提供子目标列表来定义目标。然后通过提供更深一层的子目标列表来定义子目标，等等。如果子目标的一个路径寻找不到解决方案，那么该路径就被放弃，并系统地尝试另一路径。

建立的程序可以是文本形式或可视形式。在可视化语言程序中，操作元素是图形而不是文本。

1.2 编译或解释

使用人可阅读形式的计算机程序，计算机编程语言叫作源代码。使用编译器，或者在解释器的帮助下立即执行可以将源代码转换为可执行形式。

编译的计算机程序通常被称为可执行的、二进制形式或者就是二进制——一种用二进制文件格式用来存储可执行代码的形式。用编译器把来自编程语言的源代码转换为目标代码或

机器代码。目标代码需要进一步处理才能成为机器代码，而机器代码是中央处理器自己的代码，随时可以执行。

解释的计算机程序或者是先解码然后立即执行，或者被解码为某些有效的中间形式以便以后执行。BASIC、Perl 及 Python 是立即执行的计算机程序例子。而 Java 计算机程序是预编译的并以与机器无关的叫作字节码的代码存储。然后字节码可以被叫作虚拟机的解释器按照请求执行。

解释器的主要缺点是如果被编译，计算机程序运行就比较慢。解释代码比运行其编译版本慢，因为解释器必须对每次装入的每个语句解码，然后才能执行希望的操作。另一方面，使用解释器软件开发可能更快，因为省略了编译步骤后立即进行测试。解释器的另一个缺点是在计算机程序运行时计算机必须有解释器。相反，编译的计算机程序在执行时就不需要编译器。

编程语言不需要专有的编译或解释。分类反映了最流行的语言执行方法。例如，尽管有 BASIC 编译器和 C 解释器的存在，BASIC 还是被当作一个解释语言，C 语言被当作编译语言。某些系统使用即时编译（JIT），因此源代码段被“不断地”编译并存储以备执行。

2. 执行和存储

典型情况下，计算机程序存储在非易失内存中，直到被计算机用户请求直接或间接地执行。当发出请求后，该程序被叫作操作系统的程序装入到随机访问存储器中，在那里可以被中央处理器直接访问。中央处理器然后执行（“运行”）该程序，一条指令接一条指令执行，直至结束。一个执行中的程序叫作一个进程。既可以被自身正常终止，也可以被错误（软件或硬件错误）终止。

2.1 嵌入式程序

某些计算机程序可以嵌入到硬件中。一个存储程序计算机需要存储在其只读内存中的初始化程序来引导。这个引导过程识别和初始化系统的各方面，从 CPU 寄存器到设备控制器到内存内容。初始化之后，这个初始计算机程序装载操作系统并设置程序计数器以开始常规运行。独立于主机之外，硬件设备可能由嵌入的固件来控制其运行。当很少希望或者永远不希望改变计算机程序，或者关机后程序不应该丢失时，那就使用固件。

（图略）

2.2 手工编程

过去通过开关手工地把计算机程序输入到中央处理器。一个指令由设置一组开/关配置来表示。在设置了该配置后，按下执行按钮。然后重复这一过程。过去也通过纸带或穿孔卡手工地输入计算机程序。在介质装入后，通过开关和按下执行按钮来设置开始地址。

（图略）

2.3 自动程序生成

生成编程是一种计算机编程，它通过一般类、原型、模板、外形和代码生成器产生源代码，以提高程序员的生产率。源代码用像模板处理器或集成开发环境这样的编程工具生成。源代码生成器的最简单例子就是宏处理器，如 C 语言预处理程序，它可以根据相对简单的

规则替代源代码中的样本。

软件引擎输出源代码或标识代码，该代码同时输入到另一个计算机进程中。这类似于用计算机代码作燃料，用一个进程驱动另一个进程。应用服务器是把应用软件交付给客户计算机的软件引擎。

2.4 同时执行

许多操作系统支持多任务，它可以使多个计算机程序好像同时运行在一个计算机上。操作系统也可以通过进程调度来运行多个程序 —— 一个在多个进程间频繁切换 CPU 的软件机制，这样用户可以在运行中与每个程序交互。在硬件中，现代多处理器计算机或带有多核处理器的计算机可以运行多个程序。

3. 功能分类

计算机程序也可以按照功能来分类。按照功能可以分为系统软件和应用软件。系统软件包括把计算机硬件与应用软件相结合的操作系统。操作系统的目的提供一个应用软件方便和高效运行的环境。除了操作系统外，系统软件还包括帮助人们管理和调整计算机的实用程序。如果一个计算机程序不是系统软件那就是应用软件。应用软件包括中间件，它把系统软件与用户接口相结合。应用软件也包括帮助用户解决应用问题（如需要排序）的实用程序。

Lesson 6

Text

C Language

C was developed in the early 1970s, and it has grown into a very popular language now. C might best be described as a "medium level language". Like a true high level language, there is a one-to-many relationship between a C statement and the machine language instructions it is compiled into. Thus, a language like C gives you far more programming leverage than a low level assembly language. However, compared to most high-level languages, C has a very small set of constructs. In addition, unlike most high level languages, C let you easily do chores (such as bit and pointer manipulation) additionally performed by assembly languages. Therefore, C is an especially good tool to use for developing operating systems (such as the UNIX operating system), or other system software.

The C language does not support I/O. Instead, in order to perform I/O operations, you make a system call or call a library (CLIB) function.

1. Two Ways to Call C

We only support one C compiler; however, this one compiler can be called by invoking either of the two following files:

```
/bin/cc
/com/cc
```

File /bin/cc is just a command line parser. Once it finishes a slight bit of analysis on your command line, it calls up /com/cc. File /com/cc has its own command line parser, and also contains the C preprocessor and compiler. We created two different command line parsers so that IX system users could use the traditional UNIX C compiler options and AEGIS system users could use all the traditional AEGIS compiler options.

2. Sample Program

The best way to get started with C is to write, compile, and execute a simple program. Here is a simple program to get you started:

```
/* Program name is "getting_ started" */
main ()
{
```

```
int x, y
  printf ("Enter an integer");
  scanf ("% d", &X);
  y = x * 2
  printf ("\ n% d is twice % d\ n", x, y);
}
```

3. Compiling and Executing

Suppose that you store the program in file getting_started .c. (The filename must end with the ".c" extension; therefore, you store it in getting_started.c rather than just plant getting_started.) Compiling under the AEGIS operating system produces executable object file getting_started.bin; Compiling under the IX operating system produces executable object file a.out. To run these objects, just enter the name of the file.

4. Building Blocks of C

When describing a language implementation, it is customary to describe the basic building blocks of that implementation.

4.1 Comments

A comment is any series of characters beginning with /* and ending with */. The compiler ignores all comments. In the following example, a comment follows an assignment statement:

```
average = total/number_ of_ components; /* Find the mean value * /
```

C allows comments to appear anywhere in the source file. Since the compiler interprets comments as nulls, this can result in unusual concatenations if you are not careful. For instance, the statement:

```
int x/* This is an example * /z;
becomes:
intxz;
```

Note:

In many other C compilers, placing a comment in the middle of a variable name produces an error.

4.2 Identifiers

Identifiers, also called names, can consist of the following:

- letters;
- digits;
- the dollar sign ($);
- the underscore (_).

However, the first character must be a letter or an underscore. For example, here are some legal and illegal identifiers:

```
meters                        /* Legal */
green_eggs_and_ham            /* Legal */
20_meters                     /* Illegal, because it starts with a digit */
no%#@good                     /* Illegal, because it contains illegal characters */
```

The maximum length of an identifier is 32 characters, if you specify an identifier longer than 32 characters, the C compiler truncates the name after the 32nd character and issues a warning message. For example, because of truncation, the following two identifiers will be stored identically as this_is_an_extremely_long_identi:

```
this_is_an_extremely_long_identifier
this_is_an_extremely_long_identification
```

4.3 Name Spaces

All identifiers (names) in a program fall into one of the three name spaces. Names in different name spaces never interfere with each other. You can use the same name for an entity in each of the three classes without these names affecting one another. The three name spaces are:

- Structure, Union and Enumeration Tags.
  Tag names that immediately follow these type names:
  struct, union and enum.
- Member Names.
  Names of members of a structure or union.
- All Other Names.
  Any names that are not members of the preceding four classes.

The following example uses the same name, overuse, in all three ways:

```
main ()
 {
 int overuse;           /* normal identifier */
 atruct overuse;        /* tag name */
   { float overuse;     /* member name */
   char *p;
   }
}
```

Note:

We used this example to demonstrate name spaces. However, using the same name in different name spaces can be quite confusing and is therefore not recommended.

The visibility of a variable determines whether or not the variable can be accessed in a specific region. A variable can become invisible throughout a region if another variable with the same name

and name space is declared within the region in a new block. In the following example, the assignment statement refers to the x declared within the block, not the x declared outside the block. The outer x is invisible so long as the inner x is active.

Macro names do interfere with the other four name spaces. Therefore, when you specify a macro name, do not use this name in one of the other four name spaces. For example, the following program fragment contains a macro named square and a label named square:

```
#define square (arg) arg * arg
main ()
{
 :
  square:
 :
  }
```

In the preceding example, we want square: to be treated as a label, but instead the preprocessor will treat it as if it were a macro.

4.4 Storage Class

Every variable has several characteristics. One of those characteristics is its data type. Another characteristic is its storage class which divides into two further characteristics duration and scope. You can control a variable's storage class through:

position	where in the file you declare the variable
storage class specifier	an optional keyword in a declaration

We begin by defining duration and scope, then we examine how position and storage class specifier affect duration and scope.

5. Duration of a Variable Declaration

The duration of a variable is the period of time during which storage is allocated to the variable. There are two categories of duration: dynamic and fixed. A variable with dynamic duration is created anew each time the block in which it is declared is entered. When the program leaves the block, the variable disappears. Conversely, a variable with fixed duration exists throughout the execution of the entire program.

We can summarize the differences between fixed and dynamic variables as follows:

- Fixed variables maintain their values from one block invocation to another, but dynamic variables lose their value each time the block is deactivated.
- Fixed variables get a default initialization value of zero if you do not explicitly initialize them. However, if you do not explicitly initialize a dynamic variable, the compiler will not initialize it for you.
- The run time system initializes fixed variables only once, whereas dynamic variables, if they are declared with an initializer, are re-initialized each time their block is entered.

6. Scope of a Variable Declaration

The scope of a variable is the region in the source code over which the variable is active. If a variable is active, it means that you can use it in your source code. If a variable is not active, and you attempt to use it in your source code, the compiler issues an error.

There are four types of scope: block, function, file and program. Block scope means that the variable is active from its declaration point to the end of the block in which it is declared. Function scope means that the variable is active from its declaration point to the end of the function. File scope means that the variable is active from its declaration point to the end of the file. Global scope means that the variable is active for the entire program (including all the files of source code that comprise the program).

7. Storage Class Specifiers

As mentioned earlier, you can supply an optional storage class specifier when you declare a variable. C supports the following four storage class specifiers:

auto Specifies that the variable has dynamic duration

register Directs the compiler to store the variable in a register whenever possible. This storage class is just useful suggestion to the compiler for optimization. Any variable declared with the register storage class is treated like an auto variable

static Specifies that the variable has fixed duration

extern Specifies that the variable is external; that is, the variable is defined somewhere else (e. g., in another function or file). Variables declared with extern have fixed duration

Storage class specifiers can come before or after the type specification, but by convention, they usually come before. Here are some sample declarations that contain storage class specifiers:

```
auto int i;
register short quart;
static char dog {} = {"Fenster"};
extern float f;
```

8. Declarations and Definitions

The difference between a declaration and a definition in C is subtle but important. A declaration associates a data type with an identifier but does not actually allocate any storage for it. A definition, on the other hand, actually allocates memory. For example, consider the following declarations and definitions:

```
int x;                          /* This is a definition */
static int y;                   /* This is a definition */
extern int z;                   /* This is a declaration */
typedef char LAST_NAXT [20];    /* This is a declaration */
```

If you use the storage class specifier extern, you generate a declaration. If you use a storage class specifier other than extern, or if you omit a storage class specifier, then you generate a variable definition. Using typedef generates a declaration.

The distinction between declarations and definitions is particularly important when creating global variables.

9. Global Variables

A "global variable" (also called an "external variable") is one that can be accessed by modules in different source files; that is, a global variable has global scope. There are three ways that you can create a global variables:

- Create a variable at the top level with the extern storage class specifier. This produces a variable declaration. Such a declaration cannot contain an initializer. You can access this global variable throughout the remainder of the file.
- Create a variable at the head of block with the extern storage class specifier. This produces a variable declaration. Such a declaration cannot contain an initializer. You can access this variable only within the block in which you declare it.
- Create a variable at the top level and omit a storage class specifier. This produces a variable definition. Such a definition can contain an optional intializer.

Now that we have explained how to create various global variables, we can further distinguish between definitions and declarations.

In C, every global variable can be declared zero or more times (in different files), but must be defined at least once, and may be defined more than once in different files. You cannot, however, define a global variable more than once in the same file. If you explicitly initialize a global variable in more than one file, the last initializer read by the binder is the variable's initial value at runtime. Therefore, the order in which you list files in the bind command determines the initial values of external variables. If you do not initialize a global definition, its initial value defaults to 0.

New Words

construct	[kən'strʌkt]	*vt.* 构想，概念
chore	[tʃɔː]	*n.* 困难的工作
leverage	['liːvəridʒ]	*n.* 力量，能力
invoke	[in'vəuk]	*v.* 调用
parser	['pɑːsə]	*n.* 语法
plant	[plɑːnt]	*vt.* 放置
concatenation	[kənˌkæti'neiʃən]	*n.* 连接
identifier	[ai'dentifaiə]	*n.* 标识符

underscore	[ˌʌndəˈskɔː]	*n.* 下画线
		vt. 在……之下画线
truncate	[ˈtrʌŋkeit]	*v.* 截去，把……截短
truncation	[trʌŋˈkeiʃən]	*n.* 截去，截短（名词）
identification	[aiˌdentifiˈkeiʃən]	*n.* 标识
enumeration	[iˌnjuːməˈreiʃən]	*n.* 枚举
tag	[tæg]	*n.* 标识，标记
visibility	[ˌviziˈbiliti]	*n.* 能见度
declare	[diˈklɛə]	*vt.* 说明
invisible	[inˈvizəbl]	*adj.* 看不见的，不可见的
specifier	[ˈspesifaiə]	*n.* 标识符
dynamic	[daiˈnæmik]	*adj.* 动态的
anew	[əˈnjuː]	*adv.* 再次，重新
conversely	[ˈkɔnvəːsli]	*adv.* 相反
deactivate	[diːˈæktiveit]	*vt.* 停用
initialize	[iˈniʃəlaiz]	*vt.* 初始化
scope	[skəup]	*n.* 作用域
comprise	[kəmˈprəiz]	*v.* 构成，包含，由……组成
global	[ˈgləubəl]	*adj.* 全局，全程，全球，整体
subtle	[ˈsʌtl]	*adj.* 细微的
bind	[baind]	*v.* 绑定，赋值
binder	[ˈbaində]	*n.* 联编程序

Phrases

grow into	成为
medium level language	中级语言
high level language	高级语言
low level language	低级语言
make a system call	调用系统
building block	结构块
name space	名空间
fall into	属于
tag name	标识名
one to many	一对多
head of block	块首
top level	顶层
member name	成员名

macro name	宏名称
storage class	存储类
run time system	运行（时）期系统
global scope	全局作用域
by convention	按常规
storage class specifier	存储类标识符

Notes

［1］However，compared to most high-level languages，C has a very small set of constructs.

一般地说，compared with 的意思是“将……与……比较”，强调比较两者（同类、性质相同）的不同之处。compared to 的意思是“将……比作……”，两者不同类，性质不同，但在某一点有相似之处。但有时作“将……与……比较”意思时，用 compare with/to 均可。请看下例：

Shanghai is larger compared with/to Suzhou.

本句中 compared to most high-level languages 等于 compared with most high-level languages。

［2］Instead，in order to perform I/O operations，you make a system call or call a library（CLIB）function.

本句中，make a system call 中的 call 是名词“调用”，而 call a library（CLIB）function 中的 call 是动词，意思也是“调用”。

［3］Macro names do interfere with the other four name spaces.

本句中，do 在此处是助动词，表示强调，意思是“的确”。还可以是“一定、千万”等意思。请看下例：

Do be careful，please.

Do wait for me at the gate.

［4］A variable with dynamic duration is created anew each time the block in which it is declared is entered.

本句中，介词短语 with dynamic duration 作定语，修饰和限定 a variable。in which it is declared 是一个定语从句，修饰和限定 the block。each time the block in which it is declared is entered 是一个时间状语从句，修饰 is created。

［5］The run time system initializes fixed variables only once，whereas dynamic variables，if they are declared with an initializer，are re-initialized each time their block is entered.

本句中，whereas 是一连词，意思是“but in contrast；while on the other hand”，汉语意思是“反之；而在另一方面却”；而 if 引导了一个条件状语从句。

［6］The scope of a variable is the region in the source code over which the variable is active.

本句中，in the source code over which the variable is active 作定语，修饰和限定 the region。over which the variable is active 是一个定语从句，修饰和限定 the source code。

Grammar

动　名　词

1. 动名词的定义和特征

1）动名词的定义

动名词是一种非谓语动词形式，由动词原形加词尾 ing 构成，它没有人称和数量的变化。

2）动名词的特征

动名词是动词的一种形式，故也具有动词的特征，可以带有自己的宾语和状语。请看下例：

【例】She enjoys reading books on computer science.

她喜欢读计算机方面的书籍。

句中，reading 是动名词，books on computer science 作 reading 的宾语。

【例】Excuse me for coming late.

对不起，我来晚了。

句中，coming 是动名词，late 作 coming 的状语。

由于动名词还具有名词的特征，因此在句子中可作主语、表语、宾语、介词宾语、定语、补足语等。

2. 动名词在句子中的作用

1）作主语

【例】Smoking is forbidden here.

此地禁止吸烟。

【例】Learning computer science is very important now.

现在学习计算机很重要。

【例】Learning computer well is no easy job.

学好计算机不是一件容易事。

动名词作主语时，也可用 it 作形式主语，放在句首，而将真正的主语——动名词短语放在谓语之后。请看下例：

【例】It's no use talking like that.

这样说是没有用的。

【例】It is a waste of time arguing about it.

辩论这件事是浪费时间。

【例】It's very difficult getting everything ready in such a short period of time.

要在这么短的一段时间内把一切都准备好非常困难。

2）作表语

【例】His job is developing software.

他的工作是开发软件。

【例】Seeing is believing.

眼见为实。

句中，动名词 seeing 作主语，believing 作表语。

【例】The real problem is finding out where the trouble is.

真正的问题是找出故障所在。

3）作宾语

【例】This printer needs repairing.

这台打印机需要修理一下。

句中，动名词 repairing 作 needs 的宾语。

【例】Hearing the news，she couldn't help laughing.

听到这个消息，她禁不住大笑起来。

句中，hearing the news 是现在分词短语作时间状语，动名词 laughing 作 help 的宾语。

【例】Would you mind filling out this form？

请填一下这张表，好吗？

句中，动名词短语 filling out this form 作 mind 的宾语。

【例】Have you finished writing your letter？

你把信写完了吗？

英语中，suggest、finish、avoid、stop、admit、keep、require、postpone、practise、fancy、deny 等动词都用动名词作宾语，不能用不定式作宾语。但在 love、like、hate、begin、start、continue、remember、forget、regret 等词后面可以用动名词作宾语，也可以用动词不定式作宾语。请看下例：

【例】Do you like watching/to watch TV？

你喜欢看电视吗？

【例】The man began making /to make plans for the work.

这人开始制订工作计划。

这两种结构之间有时差别不大，有时却有不同的意思。请看下例：

【例】I remember seeing her before.

我记得以前见过她。

句中，seeing her before 是动名词短语，作谓语 remember 的宾语。seeing her 的动作已经发生。

【例】You must remember to lock the door before you leave.

记住在你离开前锁门。

句中，to lock the door 是动词不定式，作谓语动词 remember 的宾语。to lock the door 的动作还没有发生，需要去做。

【例】I regret not having accepted your advice.

我后悔没有听从你的劝告。

句中，regret 后面跟动名词短语的意思是对已经发生的事感到遗憾。

【例】She didn't regret telling me what she thought.

她不后悔把她的想法告诉了我。

【例】I regret to inform you that we are unable to offer you employment.

我很遗憾地通知你，我们不能雇用你。

句中，regret 后面跟动词不定式的意思是对将要说出的话感到遗憾。

【例】She forgot posting the letter.

她忘记寄过信的事。

句中，forget 后面跟动名词表示忘记已作过或已经发生的事。

【例】She forgot to close the window when she left.

她离开时忘了关窗户。

句中，forget 后面动词不定式表示忘记要做的事。

动名词作宾语时，如本身带有补足语，则常用 it 作形式宾语。而将真正的宾语——动名词放在补足语的后面。请看下例：

【例】Do you think it necessary repeating the experiment?

你认为有必要重做这个实验吗?

句中，it 作 think 的形式宾语，真正的宾语是动名词短语 repeating the experiment。形容词 necessary 作宾语补足语。

【例】She thought it worthwhile spending so much time on that problem.

她认为在那个问题上花那么多时间是值得的。

【例】I found it useless arguing with her.

我发现与她辩论没有用。

4）作介词宾语

【例】He insisted on repairing the printer himself.

他坚持自己修理那台打印机。

【例】Have you got used to getting up early?

你习惯早起了吗?

【例】What prevented you from coming to the meeting yesterday?

何事让你昨天没来开会?

【例】Before installing the laser printer, you should read this guide first.

在安装激光打印机之前，应先阅读这个手册。

【例】Thank you for giving me so much help.

谢谢你给我那么多帮助。

5）作定语

【例】You may use these articles as reading materials.

你可以把这些文章作为阅读材料。

句中，动名词 reading 作定语，修饰 materials。

【例】English is one of the working languages at international meetings.

英语是国际会议上使用的工作语言之一。

【例】He worked in a factory that produces building materials.

他在一家生产建筑材料的工厂工作。

6）作补足语

【例】We call this process testing.

我们称这个过程为检测。

句中，动名词 testing 作宾语 this process 的补足语。

【例】This process is called testing.

这个过程被称为检测。

3. 带有名词或代词的动名词结构

动名词短语前面加物主代词或名词所有格时，可表示这个动名词的逻辑主语。这种动名词结构可在句子中作主语、表语、宾语和介词宾语。请看下例：

【例】His going there will be of great help.

他到那里去会有很大的帮助。

句中，物主代词 his 作动名词短语 going there 的逻辑主语。his going there 在句子中作主语。

【例】Mike's coming late made him angry.

迈克来晚了，这使他很生气。

句中，名词所有格 Mike's 作动名词短语 coming late 的逻辑主语。Mike's coming late 在句子中作主语。

【例】The problem is their not having enough money.

问题是他们资金不够。

句中，their 作动名词短语 not having enough money 的逻辑主语。their not having enough money 在句子中作表语。

【例】What troubles her is Peter's not having repaired the printer.

使她烦恼的是彼得还没有修理好打印机。

句中，名词所有格 Peter's 作动名词短语 not having repaired the printer 的逻辑主语。Peter's not having repaired the printer 在句子中作表语。

【例】Do you mind my opening the window？

我把窗户打开，您不介意吧？

句中，my 作 opening the window 的逻辑主语。my opening the window 在句子中作谓语动词 mind 的宾语。

【例】He thought Mike's not having learned computer a great mistake.

他认为迈克没学过计算机是个大错误。

句中，Mike's not having learned computer 作谓语动词 thought 的宾语，a great mistake 作宾语补足语。

【例】he insisted on his going there alone.

他坚持一个人到那去。

【例】He was surprised at her coming.

她的到来使他很惊奇。

4. 动名词的完成式和被动式

1）动名词的完成式

动名词的一般式通常表示一般性的动作，没有时间界限，或者表示与谓语动词所表示的动作同时发生。请看下例：

【例】He is interested in reading science fictions.

他对读科幻小说感兴趣。

【例】The fuel began burning and the temperature began going up.

燃料开始燃烧，温度便开始上升。

句中，动名词 burning 和动名词短语 going up 与 began 同时发生。

动名词的完成式表示动名词的动作在谓语动词的动作之前发生。请看下例：

【例】I regret not having told her the news.

我后悔没有告诉她这个消息。

句中，not having told her the news 发生在 regret 之前。

【例】I remember having done this experiment before.

我记得以前做过这个实验。

句中，having done this experiment before 发生在 remember 之前。

【例】He was praised for having developed a new type of machine.

他因研制了一种新型机器而受到赞扬。

应该注意的是，在某些动词或介词后可用动名词的一般式来表示在谓语动词之前发生的动作。请看下例：

【例】Thank you for helping us.

谢谢你给我们的帮助。

【例】Please excuse me for being late.

请原谅我来晚了。

【例】On hearing the news, he burst into laughter.

听到这个消息之后，他放声大笑起来。

2）动名词的被动式

当一个动名词逻辑上的主语所表示的是这个动作的对象时，这个动名词一般要用被动形式。请看下例：

【例】This question is far from being settled.

这个问题远远没有解决。

句中，being settled 是动名词的被动式，它和 This question 之间的关系是被动关系。This question 是 settled 的逻辑上的宾语，即作用的对象。

【例】They insisted on their device being tested under operating conditions.

他们坚持他们的设备要在使用条件下试验。

【例】People know of electricity having been discovered for centuries.

人们知道，电在几百年前就被发现了。

【例】He remembered the software's having been tested.

他记得这个软件已经测试过了。

值得注意的是，有时主动语态的动名词具有被动的意思。请看下例：

【例】The software needs testing.

该软件需要测试。

【例】This book is worth buying.

这本书值得买。

【例】Her design requires improving.

她的设计需要改进。

5. 动名词和现在分词的区别

（1）动名词和现在分词在句子中所作的句子成分不同。归纳如下表：

	主语	复合谓语	表语	宾语	定语	状语	补足语
动名词	是	否	是	是	是	否	是
现在分词	否	是	是	否	是	是	是

（2）动名词作表语时，表示“事实”，通常回答“干什么”；现在分词作表语时，表示“性质”，通常回答“怎么样”。请看下例：

【例】His main job is developing new products.

他的主要工作是研制新产品。

句中，developing new products 是动名词短语。

【例】An example of chemical change is breaking up water into hydrogen and oxygen.

化学变化的一个例子就是把水分解为氢气和氧气。

句中，breaking up water into hydrogen and oxygen 是动名词短语。

【例】This film is very moving.

这部影片很感人。

句中，moving 是现在分词。

（3）现在分词作定语时，表示“行为”，通常可扩展为一个定语从句；动名词作定语时，表示“用途”，通常不能扩展为一个定语从句。请看下例：

【例】building materials

建筑材料

在本词组中，building 是动名词。building materials 等于 materials for building。

【例】a sleeping car

卧车

在本词组中，sleeping 是动名词。a sleeping car 等于 a car for sleeping。

【例】a sleeping boy

一个酣睡的男孩

在本词组中，sleeping 是现在分词。a sleeping boy 可以扩展为 a boy who is sleeping。

【例】running water

自来水

在本词组中，running 是现在分词。running water 等于 water which runs。

Exercises

一、根据课文内容，判断以下叙述的正误。

（1）C is one of the most popular high-level languages nowadays.

（2）C gives the same programming leverage as assembly languages.

（3）C doesn't let you do chores performed by any assembly languages.

（4）Comments must appear at the beginning of a program.

（5）Some systems ignore comments.

（6）Comments may appear in the middle of a variable name.

（7）Data type is one of the characteristics of variables.

（8）Fixed variables maintain their value during the invocation.

（9）A global variable can only be defined once.

（10）The global variable's initial value defaults to 1.

二、根据课文内容填空。

（1）The development of C began ______________.

（2）In C environment, you ______________to perform I/O operation.

（3）The ways to call C are ______________.

（4）A comment begins with ______________and ends with ______________.

（5）Identifiers can be made up of ______________.

（6）The first character in an identifier must be ______________.

（7）The maximum length of an identifier can be ______________characters.

（8）The initial value of fixed variables is ____________when you don't explicitly initialize them.

（9）The storage class identifiers which C supports are ______________.

（10）There are ______________ways to create a global variable.

三、指出下列句子中的 V + ing 是动名词还是现在分词，并说明其在句子中的作用，然后把句子译成汉语。

（1）Selecting more than one item in a directory window is called extending a selection.

(2) Many File Manager tasks involve naming or renaming files and directories.
(3) We can not understand why he avoided speaking to us.
(4) Tom fell asleep the moment he got home, leaving the door open.
(5) Beginning some activities is to set about doing it.
(6) The boy came in without being asked.
(7) When she watched him coming over, her faced turned red.
(8) I don't feel like telling anyone about his coming back.
(9) Waiting outside, she felt tired and worried.
(10) They were deeply moved by the moving film.
(11) On hearing the news, he felt upset.
(12) There is a swimming pool in their school.
(13) It's no use asking him for help.
(14) Their living conditions were very poor several years ago.
(15) After finishing their work, they left their office.
(16) All the tickets having been sold out, we returned upset.

四、从供选择的词汇中选择最合适的填在文中相应数字处。

An expert system is a software that ____1____ specialist knowledge about a particular domain of ____2____ and is capable of making ____3____ decisions within that domain. Although expert systems typically focus on a very narrow domain, they have achieved dramatic success with ____4____ problems. This has excited widespread interest outside the research laboratories from which they emerged.

Expert systems have given rise to set of "knowledge engineering" methods constituting a new approach to design of high performance software system. This new approach represents an ____5____ change with revolutionary consequences.

供选择的答案：

A. develops
B. directory
C. effect
D. encapsulates
E. evolutionary
F. experience
G. expertise
H. intelligent
I. Real-time
J. Real-life

五、听短文，在画线处填写所听到的单词或词组。

In computing, C is a general-purpose, ____1____, block structured, procedural, ____2____

computer programming language ___3___ in 1972 by Dennis Ritchie at the ___4___ Telephone Laboratories for use with the ___5___ operating system.

Although C was designed for implementing ___6___ software, it is also widely used for developing ___7___ software.

It is widely used on a great many different software platforms and computer ___8___, and several popular ___9___ exist. C has greatly influenced many other popular programming languages, most notably C++, which originally began as an ___10___ to C.

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

1. A ________ computer is a personal computer whose hardware is capable of using any or all of the following media in a program: audio, text, graphics, video and animation.

   A. database　　B. multimedia

   C. network　　D. mainframes

2. The ________ controls the cursor or pointer on the screen and allows the user to access commands by pointing and clicking.

   A. graphics　　B. printer

   C. program　　D. mouse

3. A ________ copies a photograph, drawing or page of text into the computer.

   A. scanner　　B. printer

   C. display　　D. keyboard

4. ________ is permanently stored in the computer and provides a link between the hardware and other programs that run on the PC.

   A. Interface　　B. Operating system

   C. Internet　　D. Application software

5. ________ is not a linear structure.

   A. Graph　　B. Queue

   C. Stack　　D. I-dimension array

6. ________ is the sending and receiving of the messages by computer. It is a fast, low-cost way of communicating worldwide.

   A. LAN　　B. Post office

   C. E-Mail　　D. Interface

7. The ________ is a collection of computers connected together by phone lines that allows for the global sharing of information.

   A. interface　　B. Internet

   C. LAN　　D. WWW

8. ________ are web sites that search the web for occurrences of a specified word or phrase.

   A. Search engines　　B. WWW

   C. Internet　　D. Java

9. Files can be lost or destroyed accidentally. Keep ________copies of all data on removable storage media.

A. backup B. back

C. black D. backdown

10. In ________programming, the user determines the sequence of instructions to be executed, not the programmer.

A. top-down B. structure

C. data-driven D. event-driven

Skill Training

计算机英文论文标题写作

论文的标题具有重要意义，它是对全文的高度概括，既要便于读者通过标题知晓文章的主旨，也要便于查询。英文标题是一次创作，不是对中文标题的简单翻译。

1. 论文标题要反映文章主旨

论文标题高度概括全文内容，往往就是文章的中心论点。它具有高度的明确性，便于读者把握全文内容的核心。

标题要能够揭示论文范围或论点，使人看了标题便知晓文章的大体轮廓、所论述的主要内容以及作者的写作意图，而不能似是而非、散乱、没有重点。

标题尤其要突出论文的创新点与独特性。

由于某些论文中的内容复杂或科学术语结构复杂，难以用少量英文词写出概括全文的标题，所以国际标准化组织建议采用主标题加副标题的办法解决标题过长的问题。通过这种方法点明论文的研究对象、研究内容、研究目的，对总标题加以补充、解说。为了强调论文所研究的某个侧重面，也可以加副标题。一些商榷性的论文，一般都有一个副标题，如在总标题下方，添上“与××商榷”之类的副标题。

在写作英文标题时应用冒号将主、副标题分开，不要按中文方式用破折号将之分开。例如：Chinese and Western Modal Logic: The Difference and Its Cause（中西模态逻辑的差异及其成因）。

2. 标题要简明、精辟

论文的标题不宜过长，过长了容易使人产生烦琐和累赘的感觉，得不到鲜明的印象，从而影响对文章的总体评价。一般在 10 至 12 个单词之间，不应超过 100 个印刷符号。

标题也不能过于抽象、空洞，不能采用生僻的或自造的词汇，以免造成读者的阅读困难。

许多论文作者在中文标题中喜欢用“试论”、“初探”、“浅论”、“浅谈”、“考略”、“管窥”等表谦词语。在英文标题中表谦词语应一概删去，诸如“A Study of”、“The Exploration of”、“Research on”、“Investigation”、“The Preparation of”，以及“The Synthesis of”等。

例如，原来的标题是：Some Results of Handling Metalogical Problem of Entailment System Cm by Computer（用计算机处理制约系统 Cm 的元逻辑问题的若干结果），可将之改为：Machine-Proving of Metalogical Problems of Entailment System Cm（制约系统 Cm 的元逻辑问题的机器证明）。

3. 恰当使用缩略语

要在标题中使用行业普遍认可的缩略语，但也不要自主地随便缩写。

例如，Methodological Principles of Solving Linguistic Problems Relating to Artificial Intelligence（解决人工智能语言问题的方法论原则），可以改写为：Methodological Principles of Solving Linguistic Problems of AI。因为把 Artificial Intelligence 缩写为 AI 是行业内普遍认可的。

4. 注意字母的大小写

在论文标题中使用大写字母还是使用小写字母这一问题，并没有统一的标准，各个杂志有自己的规定。可以分为以下 3 种：

（1）只有第一个字母大写。

（2）每个单词的第一个字母大写，冠词、介词和连词除外。但是，当冠词、介词和连词为第一个单词时，其首字母也要大写。

（3）所有字母都大写。

投稿时阅读该杂志的格式要求，或查看已经出版的期刊，即可了解。

5. 标题中尽量多用关键词语

尽量在标题中使用论文中的关键词语，可增加论文的被检次数，从而可能增加被引次数。因为用机器检索时，机器只显示标题中的关键词语而不是整个标题。就此而言，标题中关键词语的使用问题应该引起重视。

Reading Material

Object-Oriented Programming[1]

Object-oriented programming（OOP）is a programming paradigm that uses "objects[2]" and their interactions to design applications and computer programs. Programming techniques may include features such as encapsulation[3], modularity, polymorphism[4] and inheritance. It was not commonly used in mainstream software application development until the early 1990s. Many modern programming languages now support OOP.

---

［1］Object-Oriented Programming（OOP）面向对象编程
［2］object［ˈɔbdʒikt］*n.* 对象
［3］encapsulation［inˌkæpsjuˈleiʃən］*n.* 封装
［4］polymorphism［ˌpɔliˈmɔːfizəm］*n.* 多态性

1. Introduction

Object-oriented programming can trace its roots to the 1960s. As hardware and software became increasingly complex, researchers studied ways in which software quality could be maintained. Object-oriented programming was deployed in part as an attempt to address this problem by strongly emphasizing discrete[1] units of programming logic and reusability in software.

The Simula programming language was the first to introduce the concepts underlying object-oriented programming (objects, classes[2], subclasses[3], virtual methods, coroutines[4], garbage[5] collection and discrete event simulation) as a superset[6] of Algol. Simula was used for physical modeling, such as models to study and improve the movement of ships and their content through cargo ports. Smalltalk was the first programming language to be called "object-oriented".

Object-oriented programming may be seen as a collection of cooperating[7] objects, as opposed to a traditional view in which a program may be seen as a group of tasks to compute ("subroutines[8]"). In OOP, each object is capable of receiving messages, processing data and sending messages to other objects.

Each object can be viewed as an independent little machine with a distinct role or responsibility. The actions or "operators[9]" on the objects are closely associated with the object. For example, in OOP, the data structures tend to carry their own operators around with them (or at least "inherit" them from a similar object or "class"). The traditional approach tends to view and consider data and behavior separately.

2. Fundamental concepts

A survey by Deborah J Armstrong of nearly 40 years of computing literature identified a number of "quarks", or fundamental concepts, found in the strong majority of definitions of OOP. They are the following:

2.1 Class

It defines the abstract characteristics of a thing (object), including the thing's characteristics (its attributes, fields or properties) and the thing's behaviors (the things it can do, or methods, operations or features). One might say that a class is a blueprint[10] or factory that describes the

---

[1] discrete [dis'kriːt] *adj.* 不连续的，离散的
[2] class [klɑːs] *n.* 类
[3] subclass ['sʌbklɑːs] *n.* 子类，子集
[4] coroutine [ˌkəruː'tiːn] *n.* 协同程序，联立程序
[5] garbage ['gɑːbidʒ] *n.* 垃圾，废物，无用信息
[6] superset ['sjuːpəset] *n.* 超集，扩展集，父集
[7] cooperating [kəu'ɔpəreitiŋ] *adj.* 协同操作的
[8] subroutine [ˌsʌbruː'tiːn] *n.* 子程序
[9] operator ['ɔpəreitə] *n.* 运算符号
[10] blueprint ['bluːˌprint] *n.* 蓝图，设计图，计划

nature of something. For example, the class Dog would consist of traits[1] shared by all dogs, such as breed and fur color (characteristics), and the ability to bark and sit (behaviors). Classes provide modularity and structure in an object-oriented computer program. A class should typically be recognizable[2] to a non-programmer familiar with the problem domain, meaning that the characteristics of the class should make sense in context. Also, the code for a class should be relatively self-contained (generally using encapsulation). Collectively, the properties and methods defined by a class are called members.

2.2 Object

It is a particular object. The class of Dog defines all possible dogs by listing the characteristics and behaviors they can have; the object Lassie is one particular dog, with particular versions of the characteristics. A Dog has fur; Lassie has brown-and-white fur.

2.3 Instance

One can have an instance of a class or a particular object. The instance is the actual object created at runtime. In programmer jargon[3], the Lassie object is an instance of the Dog class. The set of values of the attributes of a particular object is called its state[4]. The object consists of state and the behaviour that's defined in the object's class.

2.4 Method

An object's abilities. In language, methods are verbs. Lassie, being a Dog, has the ability to bark. So bark () is one of Lassie's methods. She may have other methods as well, for example sit () or eat () or walk (). Within the program, using a method usually affects only one particular object; all Dogs can bark, but you need only one particular dog to do the barking.

2.5 Message passing

Message passing is "The process by which an object sends data to another object or asks the other object to invoke a method." It is also known to some programming languages as interfacing. For example, the object called Breeder may tell the Lassie object to sit by passing a "sit" message which invokes Lassie's "sit" method. The syntax varies between languages, for example: in Objective-C. In Java code-level message passing corresponds to "method calling[5]".

2.6 Inheritance

"Subclasses" are more specialized versions of a class, which inherit attributes and behaviors

[1] trait [treit] *n.* 特性，特点
[2] recognizable [ˈrekəgnaizəbl] *adj.* 可认识的，可辨认的
[3] jargon [ˈdʒɑːgən] *n.* 行话
[4] state [steit] *n.* 情形，状态
[5] calling [ˈkɔːliŋ] *n.* 调用

from their parent classes, and can introduce their own.

For example, the class Dog might have sub-classes called Collie[1], Chihuahua[2], and GoldenRetriever. In this case, Lassie would be an instance of the Collie subclass. Suppose the Dog class defines a method called bark ( ) and a property called furColor. Each of its sub-classes (Collie, Chihuahua and GoldenRetriever) will inherit these members, meaning that the programmer only needs to write the code for them once.

Each subclass can alter its inherited traits. For example, the Collie class might specify that the default[3] furColor for a collie is brown-and-white. The Chihuahua subclass might specify that the bark ( ) method produces a high pitch by default. Subclasses can also add new members. The Chihuahua subclass could add a method called tremble ( ). So an individual chihuahua instance would use a high-pitched bark ( ) from the Chihuahua subclass, which in turn inherited the usual bark ( ) from Dog. The chihuahua object would also have the tremble ( ) method, but Lassie would not, because she is a Collie, not a Chihuahua. In fact, inheritance is an is-a' relationship[4]: Lassie is a Collie. A Collie is a Dog. Thus, Lassie inherits the methods of both Collies and Dogs.

Multiple inheritance is inheritance from more than one ancestor class, neither of these ancestors being an ancestor of the other. For example, independent classes could define Dogs and Cats, and a Chimera object could be created from these two which inherits all the (multiple) behavior of Cats and Dogs. This is not always supported, as it can be hard both to implement and to use well.

2.7 Encapsulation

Encapsulation conceals[5] the functional details of a class from objects that send messages to it. For example, the Dog class has a bark ( ) method. The code for the bark ( ) method defines exactly how a bark happens (e.g., by inhale ( ) and then exhale ( ), at a particular pitch and volume). Timmy, Lassie's friend, however, does not need to know exactly how she barks. Encapsulation is achieved by specifying which classes may use the members of an object. The result is that each object exposes[6] to any class a certain interface—those members accessible to that class. The reason for encapsulation is to prevent clients of an interface from depending on those parts of the implementation that are likely to change in future, thereby allowing those changes to be made more easily, that is, without changes to clients. For example, an interface can ensure that puppies[7] can only be added to an object of the class Dog by code in that class. Members are often specified as public, protected or private[8], determining whether they are available to all classes, sub-classes or only the defining class. Some languages go further: Java uses the default access modifier to restrict

---

[1] Collie [ˈkɔli] *n.* 柯利牧羊犬
[2] Chihuahua [tʃiˈwɑːwə] *n.* 吉娃娃（一种墨西哥狗）
[3] default [diˈfɔːlt] *n.* 默认（值）
[4] relationship [riˈleiʃənʃip] *n.* 关系，关联
[5] conceal [kənˈsiːl] *vt.* 隐藏，隐蔽
[6] expose [iksˈpəuz] *vt.* 使暴露，受到
[7] puppy [ˈpʌpi] *n.* 小狗，幼犬
[8] private [ˈpraivit] *adj.* 私有的

access also to classes in the same package, C# and VB. NET reserve[1] some members to classes in the same assembly using keywords internal (C#) or Friend (VB. NET), and Eiffel and C++ allows one to specify which classes may access any member.

2.8 Abstraction[2]

Abstraction is simplifying complex reality by modelling classes appropriate to the problem, and working at the most appropriate level of inheritance for a given aspect of the problem.

For example, the Dog Lassie may be treated as a Dog much of the time, a Collie when necessary to access Collie-specific attributes or behaviors, and as an Animal (perhaps the parent class of Dog) when counting Timmy's pets[3].

Abstraction is also achieved through Composition. For example, a class Carwould be made up of an Engine, Gearbox, Steering objects, and many more components. To build the Car class, one does not need to know how the different components work internally, but only how to interface with them, i. e., send messages to them, receive messages from them, and perhaps make the different objects composing the class interact with each other.

2.9 Polymorphism

Polymorphism allows you to treat derived class members just like their parent class' members. More precisely, Polymorphism in object-oriented programming is the ability of objects belonging[4] to different data types to respond to method calls of methods of the same name, each one according to an appropriate type-specific behavior. One method, or an operator such as +, -, or *, can be abstractly applied in many different situations. If a Dog is commanded to speak (), this may elicit a bark (). However, if a Pig is commanded to speak (), this may elicit an oink (). They both inherit speak () from Animal, but their derived class methods override[5] the methods of the parent class; this is Overriding Polymorphism. Overloading Polymorphism is the use of one method signature, or one operator such as "+", to perform several different functions depending on the implementation. The "+" operator, for example, may be used to perform integer addition, float[6] addition, list concatenation, or string concatenation. Any two subclasses of Number, such as Integer and Double, are expected to add together properly in an OOP language. The language must therefore overload the concatenation operator, "+", to work this way. This helps improve code readability[7]. How this is implemented varies from language to language, but most OOP languages support at least some level of overloading[8] polymorphism. Many OOP languages also support Parametric

---

[1] reserve [ri'zə:v] *vt.* 储备，保存，保留

[2] abstraction [æb'strækʃən] *n.* 抽象，提取

[3] pet [pet] *n.* 动物

[4] belonging [bi'lɔŋiŋ] *n.* 附属品，附件；属性

[5] override [əuvə'raid] *vt.* 不顾，无视

[6] float [fləut] *n.* 浮点
vi. 浮动

[7] readability [ˌri:də'biliti] *n.* 易读，可读性

[8] overload ['əuvə'ləud] *vt.* 重载

Polymorphism, where code is written without mention of any specific type and thus can be used transparently with any number of new types. Pointers are an example of a simple polymorphic routine that can be used with many different types of objects.

Not all of the above concepts are to be found in all object-oriented programming languages, and so object-oriented programming that uses classes is called sometimes class-based programming. In particular, prototype-based programming does not typically use classes. As a result, a significantly different yet analogous terminology is used to define the concepts of object and instance, although there are no objects in these languages.

参 考 译 文

C 语 言

C 语言的开发始于 20 世纪 70 年代初期，现在已经成为非常流行的语言。C 语言也许最合适被称为“中级语言”。像真正的高级语言一样，一个 C 语句与编译到机器上的语言指令的关系是一对多的关系。因此，像 C 语言这样的编程手段远远超过低级的汇编语言。然而，与大多数高级语言相比，C 语言的结构非常小。另外，与大多数高级语言不同，C 语言使操作者很容易地做由汇编语言执行的工作（如操作位与指针）。因此，C 语言是开发操作系统（如 UNIX 操作系统）或其他系统软件特别好的工具。

C 语言不支持 I/O 接口。若要执行 I/O 操作，则可进行系统调用或调用一个库（CLIB）函数。

1. 调用 C 语言的两种方法

我们只支持一个 C 编译程序；然而，通过请求下列文件之一可以调用这个编译程序：

```
/bin/cc
/com/cc
```

文件/bin/cc 只是一个命令行语法。该文件一旦在命令行上略作分析，就调用/com/cc 文件。文件/com/cc 有它自己的命令行语法，也包含 C 语言预处理器和编译程序。我们建立了两个不同的命令行语法，以便 IX 系统用户能够使用传统的 UNIX C 编译程序选项，AEGIS 系统用户可以使用所有传统的 AEGIS 编译程序选项。

2. 程序范例

开始 C 语言的最好方法是编写、编译和执行一个简单程序。如下为一简单程序：
（程序略）

3. 编译和执行

假定要把程序存入 getting_started.c 中（文件必须以扩展名为“.c”结束；因此，把该程序存放到 getting_started.c 文件中而不是 getting_started 文件中）。在 AEGIS 操作系统下编译能

产生可执行的目标文件 getting_started.bin。在 IX 操作系统下编译能产生可执行的目标文件 a.out。要运行这些目标文件，只需输入文件名。

4. C 语言结构块

叙述一种语言的执行过程时，习惯于叙述该执行过程的基本结构块。

4.1　注释

一个注释就是任一系列的以/＊开头和以＊/结尾的字符。编译程序忽略全部注释。在下面的例子中，一个注释跟在赋值语句的后面：

```
average = total /number of components; /* Find the mean value * /
```

C 语言允许注释出现在源文件中的任何地方。既然编译程序认为注释无效，若不小心，就会导致异常连接。例如：

```
int x/* This is an example * /z;
变成：
int xz;
```

注意：在许多其他 C 语言程序中，把一个注释放在变量名中间会产生错误。

4.2　标识符

标识符，也可以称为名字，可由下列部分组成：

- 字母；
- 数字；
- 美元符（$）；
- 下画线（_）.

然而，第一个字符必须是一个字母或一个下画线。下面举例说明一些合法的和非法的标识符：

```
meters                    /* 合法 * /
green eggs and ham        /* 合法 * /
20_ meters                /* 非法，因为以数字开始 * /
no% #@ good               /* 非法，因为含有非法字符 * /
```

一个标识符最长可有 32 个字符。若指定的标识符超过 32 个字符，C 语言编译程序截去第 32 个以后的字符，并发出警告信息。例如，由于截去一些名字，下面两个标识符将以 this_ is_ an_ extremely_ long_ identi 形式存储：

```
this_ is_ an_ extremely_ long_ identifier
this_ is_ an_ extremely_ long_ identification
```

4.3　名空间

一个程序中所有的标识符（名字）可属于三个名空间之一。不同的名空间中的名称永

远不会互相干扰。一个程序的三种名称可以是相同的，这些名称不会互相影响。这三个名空间是：

- 结构、联合和枚举标识。

标识名紧跟在下列名称之后：

struct、union 及 enum。

- 成员名。

一个结构或联合成员中的名称。

- 所有其他名。

除了以上四种类型的成员的名称之外的任何名称。

下面的例子在所有这三个方面都使用了相同的名字 —— overuse：

```
main()
{
int overuse;          /* 一般标识符 */
atruct overuse;       /* 标识名 */
   {float overuse;    /* 成员名 */
   char *p;
   }
}
```

注意：用以上例子来说明名空间。但是，在不同的名空间中使用相同的名称会使人颇感混乱。因此，建议不要这样做。

一个变量的能见度决定该变量是否能在某一特定区域被存取。若在一个新块的区域内说明了一个相同名字和名空间的变量，则此变量就在整个区域外不可见。在下列例子中，赋值语句指的是块内 x 而不是块外的 x。只要块内的 x 是活动的，块外的 x 就不可见。

宏名称的确干扰其他四个名空间。因此，指定一个宏名称时，不要再在其他四个名称中使用这个名称。例如，下面一段程序含有一个名为“square”的宏和一个名为“square”的标识符：

（程序略）

在以上例子中，若让“square:”作为标记，但是预处理器却将它作为一个宏。

4.4 存储类

每个变量都有几个特征。其中之一是其数据类型。另一个特征是其存储类。其存储类可以进一步分成两个特征：持续时间和作用域。可通过下列两种方法控制存储类：

位置	在文件中说明变量的地点
存储类标识符	声明中的一个可选的关键词

先定义持续时间和作用域，然后查看位置和存储类标识符是如何影响持续时间和作用域的。

5. 变量说明的持续时间

变量的持续时间就是分配给变量存储的这段时间。持续时间有两种：动态的持续时间和

固定的持续时间。每当程序进入变量说明块时，一个动态持续变量就会重新建立。程序离开该块时，该变量消失，相反，一个固定持续变量在执行整个程序的过程中一直存在。

可以把固定变量和动态变量的区别总结如下：

- 固定变量在从一个块到另一个块的调用过程中其值不变，而动态变量在每次停用该块时其值消失。
- 若不明确地初始化固定变量，则固定变量得到的默认值为零。然而，若不明确地初始化动态变量，则编译程序将不替操作者初始化动态变量。
- 运行期间系统只初始化一次固定变量，而对于动态变量，若用初始程序说明，则每当进入动态变量块时，就重新初始化。

6. 变量说明作用域

变量作用域就是变量在源代码中有效的区域。若变量有效，则意味着可在源代码中使用该变量。若变量无效，却试图在源代码中使用该变量，则编译程序报告出错。

作用域有四种类型：块、函数、文件和程序。块作用域是指从变量说明点到块结束之间，变量有效。函数作用域是指从变量说明点到函数结束之间，变量有效。文件作用域是指从变量说明点到文件结束之间，变量有效。全局作用域指对整个程序（包括构成该程序的源代码的所有文件）变量都有效。

7. 存储类标识符

如上所述，说明一个变量时，可提供一个可选的存储类标识符。C 语言支持以下四种存储类标识符：

auto	指定变量有动态持续时间
register	只要可能，命令编译程序把变量存入寄存器中。这个存储类对编译程序进行优化非常有用。任何用“register”存储类说明的变量都被看作自动变量
static	指定变量有固定持续时间
extern	指定变量是外部的，即量被定义在别的地方（例如，在另一个函数或文件中）。用“extern”说明的变量具有固定持续时间

存储类标识符可以出现在类型标识之前或之后。但是，按常规它们通常出现在前面。下面是含有存储类标识符的说明范例：

（程序略）

8. 说明和定义

在 C 语言中，说明和定义之间的区别很细微，但是非常重要。说明把数据类型和标识符联系起来，但却不给标识符分配存储。而定义却实际上分配内存。例如，假定有如下说明和定义：

```
int x;                          /*这是一个定义*/
static int y;                   /*这是一个定义*/
extern int z;                   /*这是一个说明*/
typedef char LAST NAXT [20];    /*这是一个说明*/
```

若使用存储标识符“extern”，则产生说明。如果使用除“extern”以外的存储标识符，或者省略存储标识符，那么产生变量定义。使用“typedef”产生说明。

建立全局变量时，说明和定义之间的区别特别重要。

9. 全局变量

全局变量（也称为“外部变量”）是不同源文件中的模块可以访问的变量，即全局变量具有全局工作域。下面是建立全局工作域的三种方法：

- 使用存储类标识符“extern”在顶层建立一个变量。这就产生了一个变量说明。一个这样的说明不能包括初始程序，可在文件的其他部分访问此变量。
- 使用存储类标识符“extern”在块首建立一个变量。这就产生了一个变量说明。一个这样的说明不能包括初始程序。只能在块中访问此变量。
- 在顶层建立一个变量并省略存储类标识符。这就产生了一个变量定义。一个这样的定义可包含一个可选的初始程序。

既然已经解释了如何建立各种全局变量，则可进一步区别定义和说明。

在 C 语言中，每个全局变量都可说明零次或多次（在不同文件中），但是必须至少定义一次，并可以在不同的文件中定义多次。然而，不能在同一文件中多次定义一个全局变量。若在多个文件中明确地把变量初始化，则联编程序所读的最后一个初始程序是变量运行期的初始值。因此，赋值命令中文件的排列顺序决定外部变量的初始值。若没有初始化全局变量，则它的初始值默认为零。

Lesson 7

Text

About the Java Technology

Java technology is both a programming language and a platform.

1. The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

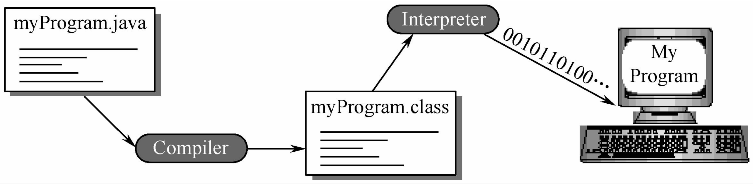
- Simple;
- Architecture neutral;
- Object oriented;
- Portable;
- Distributed;
- High performance;
- Interpreted;
- Multithreaded;
- Robust;
- Dynamic;
- Secure.

Each of the preceding buzzwords is explained in *The Java Language Environmen*, a white paper written by James Gosling and Henry McGilton.

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java bytecodes the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java bytecode instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

You can think of Java bytecodes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM.

Java bytecodes help make "write once, run anywhere" possible. You can compile your program into bytecodes on any platform that has a Java compiler. The bytecodes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



2. The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

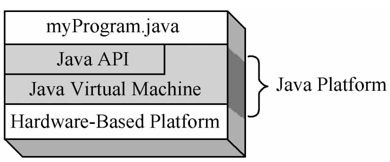
The Java platform has two components:

- The Java Virtual Machine (Java VM);
- The Java Application Programming Interface (Java API).

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time bytecode compilers can bring performance close to that of native code without threatening portability.

3. What Can Java Technology Do?

The most common types of programs written in the Java programming language are applets and

applications. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser. At the beginning of this trail is an applet that displays an animation of the Java technology's mascot, Duke, waving at you.

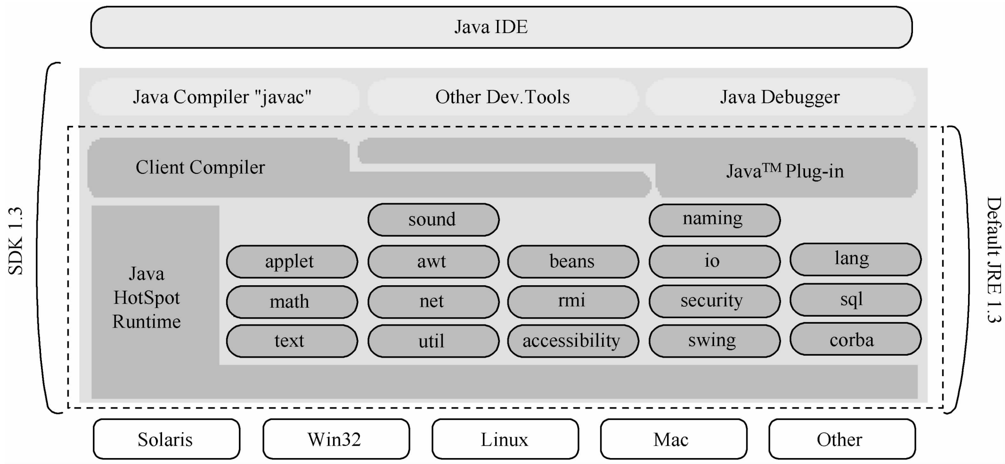
However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a server serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers and print servers. Another specialized program is a servlet. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provide a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- The essentials: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- Applets: The set of conventions used by applets.
- Networking: URL (Uniform Resource Locator), TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.
- Internationalization: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- Security: Both low level and high level, including electronic signatures, public and private key management, access control and certificates.
- Software components: Known as JavaBeans™, can plug into existing component architectures.
- Object serialization: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- Java Database Connectivity (JDBC™): Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation and more. The following figure depicts what is included in the Java 2 SDK (Software Development Kit).



Note: The Java 2 SDK, Standard Edition V. 1. 3. The Java 2 Runtime Environment (JRE) consists of the virtual machine, the Java platform core classes, and supporting files. The Java 2 SDK includes the JRE and development tools such as compilers and debuggers.

New Words

characterize	[ˈkæriktəraiz]	*vt.* 表现……的特色，具有……特征
buzzword	[ˈbʌzwəːd]	*n.* 专门用语
neutral	[ˈnjuːtrəl]	*n.* 中立者
		adj. 中立的
multithread	[ˈmʌltiθred]	*n.* 多线程
robust	[rəuˈbʌst]	*adj.* 强壮的，健壮的
interpret	[inˈtəːprit]	*v.* 解释，说明
intermediate	[ˌintəˈmiːdjət]	*adj.* 中间的
		n. 媒介
illustrate	[ˈiləstreit]	*vt.* 举例说明，图解
interpreter	[inˈtəːpritə]	*n.* 解释程序，解释器
parse	[pɑːz]	*vt.* 解析
compilation	[ˌkɔmpiˈleiʃən]	*n.* 编译；编辑
interpretation	[inˌtəːpriˈteiʃən]	*n.* 解释；说明；翻译
figure	[ˈfigə]	*n.* 图表
implementation	[ˌimplimenˈteiʃən]	*n.* 执行，履行；落实
workstation	[ˈwəːksteiʃən]	*n.* 工作站
library	[ˈlaibrəri]	*n.* 库
depict	[diˈpikt]	*vt.* 描述，描写
insulate	[ˈinsjuleit]	*vt.* 使绝缘，隔离

threaten	[ˈθretn]	*vt.* 危险，威胁
surf	[səːf]	*vi.* 冲浪
portability	[ˌpɔːtəˈbiləti]	*n.* 移植
animation	[æniˈmeiʃən]	*n.* 动画
mascot	[ˈmæskət]	*n.* 吉祥物
cute	[kjuːt]	*adj.* 可爱的，娇小可爱的，漂亮的
entertaining	[entəːˈteiniŋ]	*adj.* 愉快的，有趣的
generous	[ˈdʒenərəs]	*adj.* 丰盛的，大量的
interactive	[ˌintərˈæktiv]	*adj.* 交互式的，人机对话的
tailor	[ˈteilə]	*n.* 裁缝 *vt.* 剪裁，适应，适合
string	[striŋ]	*n.* 串
thread	[θred]	*n.* 线程
localize	[ˈləukəlaiz]	*v.* （使）局部化，（使）地方化
worldwide	[ˈwəːldwaid]	*adj.* 遍及全世界的，世界范围的；世界性的
appropriate	[əˈprəupriit]	*adj.* 适当的
signature	[ˈsignitʃə]	*n.* 签名
management	[ˈmænidʒmənt]	*n.* 管理；经营
certificate	[səˈtifikit]	*n.* 证书 *vt.* 发给证书
lightweight	[ˈlaitweit]	*adj.* 平均质量以下的
persistence	[pəˈsistəns]	*n.* 坚持，持续
uniform	[ˈjuːnifɔːm]	*adj.* 统一的，相同的，一致的
telephony	[tiˈlefəni]	*n.* 电话；电话学；电话技术
debugger	[diːˈbʌgə]	*n.* 调试器

Phrases

high-level language	高级语言
white paper	白皮书
in that	因为
translate…into…	把……翻译为……
development tool	开发工具
Web browser	Web 浏览器
as long as	只要，在……的时候
differ from	不同于
describe as	描述为
native code	本地代码
platform-independent environment	平台独立环境

close to	接近于；在附近
adhere to	坚持，追随，黏附，黏着
wave at sb.	向某人挥手
software platform	软件平台
proxy server	代理服务器
be similar to	熟悉
instead of	代替，而不是……
adapt to	适应
electronic signature	电子签名
public key	公共钥匙
private key	私密钥匙
a wide range of	各种，许多
relational database	相关数据库；关系数据库

Abbreviations

VM（Virtual Machine）	虚拟机
GUI（graphical user interface）	图形用户界面
URL（Uniform Resource Locator）	统一资源定位符
TCP（Transmission Control Protocol）	传输控制协议
UDP（User Datagram Protocol）	用户数据报协议
IP（Internet Protocol）	网际协议
JDBC（Java Database Connectivity）	Java 数据库连接
SDK（Software Development Kit）	软件开发工具包

Notes

［1］However, smart compilers, well-tuned interpreters, and just-in-time bytecode compilers can bring performance close to that of native code without threatening portability.

本句中，that 指代 performance，close to 的意思是“接近于，接近”。

［2］How does the API support all these kinds of programs? It does so with packages of software components that provide a wide range of functionality.

本句中，it 指代它前面的句子中的 the API，does so 指 support all these kinds of programs，that provide a wide range of functionality 是一个定语从句，修饰和限定 packages of software components。

本句可以改写为：

The API supports all these kinds of programs with packages of software components that provide a wide range of functionality.

Grammar

倒　装　句

1. 概述

英语中，正常语序的排列是：主语 + 谓语 + 其他句子成分。但是，因为某种句型的要求（如疑问句、there be 句型、祝愿句）或修饰的要求（如为了强调、生动、衔接上下文及平衡句子），则往往使用倒装结构的句子。

倒装句可分为全部倒装和部分倒装。在全部倒装的句子中，整个谓语都放在主语的前面。请看下例：

【例】Here is a letter for you.

这有你一封信。

【例】Now comes my turn.

现在该轮到我了。

在部分倒装句中，谓语中的一部分（如助动词、情态动词或系动词 be 等）放在主语之前，其余部分仍在主语之后。请看下例：

【例】Look! There he comes!

瞧！他来了！

【例】I couldn' t solve this problem. Nor could anyone else in our office.

我解决不了这个问题，我们办公室其他人也解决不了。

2. 常用倒装句的几种情况

（1）疑问句多数是倒装语序。

【例】Do you think it necessary to learn computer?

你认为学习计算机有必要吗?

【例】Can every substance exist in all the three states?

所有物质都能以这三态形式存在吗?

【例】Is this book worth reading?

这本书值得一读吗?

【例】What is the difference between the two?

这两者之间的区别是什么?

【例】Where did you get your computer repaired?

你在哪进修的计算机?

【例】When will the fair be?

博览会什么时候举行?

【例】How did you come here?

你怎么到这来的?

但当疑问词是主语或主语的定语时，则不倒装。请看下例：

【例】Who designed the new machine?

谁设计的这台新机器?

【例】How many people attended the party?

多少人参加了聚会?

【例】What impressed you most?

给你印象最深的是什么?

【例】Which model is the best?

哪个型号最好?

（2） 由 there、here、now、then 等引导的倒装语序。请看下例：

【例】There are many kinds of books in the library.

图书馆里有许多书。

【例】There remains one more problem to be solved.

还有一个问题要解决。

【例】There stands a new factory at the foot of the mountain.

山脚下有一个新工厂。

【例】New comes your turn.

该轮到你了。

【例】Then came a new difficulty.

然后产生了一个新的困难。

【例】Then followed three days of heavy rain.

接着下了三天大雨。

【例】Here are some models for you to choose.

这里有一些型号供你选择。

【例】Here is the most popular book of this year.

这是今年最畅销的书。

但如主语过短，则用部分倒装。请看下例：

【例】Here we are. This is the place where we will buy a printer.

我们到了。这就是我们要买打印机的地方。

【例】——May I have a look at your price list?

——Here you are.

——我能否看一下你们的报价单?

——给你。

（3） 直接引语的全部或部分放在句首，特别是引述动词较短时，引述动词和它的主语有时倒装。请看下例：

【例】"He must have gone out." thought Betty.

“他一定是出去了,” 贝蒂想到。

【例】"The book," said the boy, "is the most interesting one I've ever read."

“这本书,” 那个男孩说道,“是我读过的最有趣的书。”

但引述动词的主语是代词、或者谓语部分比主语长、或者引述部分后面还有间接宾语时，一般不用倒装。请看下例：

【例】"What is your opinion?" I said.

“你的意见呢?” 我问。

【例】"This kind of printer is very advanced. " He told me.

“这款打印机很先进。” 他告诉我。

（4）用在以 so 开头表示前面所述的情况也适用于另一个人或事物的句子，表示“……也一样”、“……也这样”。前面的句子必须是肯定句。也用在以 neither、nor 引导表示前面所述的情况也适用于另一个人或事物的句子，此时表示“……也不这样”。

【例】He has learn AutoCAD. So has Tom.

他学过 AutoCAD，汤姆也学过。

【例】Production is going up, so are the people's living conditions.

生产在提高，人们的生活条件也在提高。

【例】He can't find out where the trouble is. Nor can she.

他找不出故障所在，她也找不出。

注意：neither 与 nor 的意思相同，可以互换。no more 表示动作的程度，意思为“一样不……”。

但如果后面 so 只是单纯的重复前面句子的意思，而不表示也适用于另一人或事物，则该句不需要倒装。请看下例：

【例】"He is quite up to that work. " "So he is. "

“他很胜任那项工作。”“是的，他很胜任。”

【例】"It's cold today. " "So it is. "

“今天天气很冷。”“是挺冷。”

（5）如果虚拟语气中的条件句中省略了连词 if，则需用倒装句。请看下例：

【例】Were I in your shoes, I wouldn't give it up.

若我处在你的位置，我就不会放弃它。

句中，若不省略 if，则该句应为：

If I were in your shoes, I wouldn't give it up.

【例】Had they not been working so hard, they wouldn't have completed the project in such a short time.

如果他们不那么努力，就不会在这么短的时间内完成那项工程。

句中，若不省略 if，则该句应为：

If they had not been working so hard, they wouldn't have completed the project in such a short time.

（6）在正式文件中，as 引导的让步状语从句需用倒装语序。请看下例：

【例】Young as he is, he is very capable.

尽管他很年轻，但极有才能。

【例】Try as we might, we couldn't solve the project at once.

不管我们怎样努力，还是不能马上解决这个问题。

【例】Hard as he works, he has made little progress.

尽管他工作很努力，但进步不大。

【例】Child as he is, he knows a lot.

虽然他还是个孩子，但他懂得很多。

当表语是单数可数名词时，放在句首倒装应去掉不定冠词。

（7）并列连词 not only … but also…、neither…nor 置于句首时，句子要倒装。请看下例：

【例】Not only should you be afraid of difficulties, but you should try your best to overcome them.

你不但不应该害怕困难，而且应该竭尽全力去克服它们。

【例】Not only did he like music, but also sports.

他不但喜欢音乐，而且还喜欢体育。

【例】Neither can he do it, nor he wants to.

他做不了这件事，也不愿做。

【例】Neither has he learned computer, nor I.

他没有学过计算机，我也没学过。

但以上两个并列连词连接并列主语时，句子不倒装。请看下例：

【例】Not only he but also you are wrong.

不但他错了，你也错了。

【例】Neither he nor I can work out this problem.

他和我都算不出这道题。

（8）以否定副词 never、hardly、scarcely、not only、nor、seldom、little、rarely、nowhere 开头，或者以否定副词 by no means、at no time、not until、in no way、hardly … when …、no sooner … than …、scarcely … before（when）…开头的句子要倒装。请看下例：

【例】Never before have I met him.

我从来都没有见过他。

【例】Never has he been so glad.

他从来都没这么高兴过。

【例】Hardly is it possible to increase the speed of this car.

几乎不能再提高这辆车的速度了。

【例】Not until quite recently did he know what he should do.

直到最近他才知道他该做什么。

【例】No sooner had they got home than it started to rain.

他们刚到家，天就开始下雨了。

【例】Hardly did I know what to do next.

我几乎不知道该怎么办?

【例】At no time is smoking permitted in the meeting room.

会议室内严禁吸烟。

【例】Scarcely had she fallen asleep when the telephone rang.

她刚要睡着忽然电话铃响了。

（9）当表示频度的状语 often、once、always、many a time、now and again、seldom，方式状语 thus，以及程度状语 so 放在句首时，常用倒装句。请看下例：

【例】Many a time has he solved problems for us.

他已多次为我们解决难题。

【例】Often does his mother warn him against smoking.

他母亲经常警告他不许吸烟。

【例】Seldom do I go to the cinema.

我很少去看电影。

【例】Thus was the man cheated.

就这样，这个人被骗了。

【例】So busy is he that he has no time to watch TV.

他太忙，没时间看电视。

【例】So fast does light travel that it is difficult to imagine its speed.

光传播得那么快，以至于我们很难想象其速度。

注意：当程度状语 so 放在句首时，它所修饰的形容词和副词必须紧随其后。

（10）句首由 only 和它所修饰的状语共同充当时，必须使用倒装语序。请看下例：

【例】Only then did he realize the importance of English.

只有那时他才认识到英语的重要性。

【例】Only in this way can you improve your English.

只有通过这种方法才能提高你的英语水平。

【例】Only when you grow up can you understand it.

只有当你长大成人才能明白这件事。

但是，若 only 和它所修饰的状语不在句首时，则不倒装。若放在句首的 only 修饰的是句子的主语时，也不倒装。请看下例：

【例】You can understand it only when you grow up.

只有当你长大成人才能明白这件事。

【例】Only he was late for the meeting yesterday.

昨天只有他开会迟到了。

【例】It was reported only five men were injured in the accident.

据报导只有五个人在这次事故中受伤。

（11）为使句子更生动，可以把 up、down、in、out、back、over、off、away 等副词或副词短语以及作地点状语的介词短语放在句首，然后使用倒装语序。请看下例：

【例】The door opened and in came a girl in red.

门一开，走进一位红衣少女。

【例】In the corner stands a table.

墙角有一张桌子。

【例】Out rushed the boy.

那男孩一下子冲了出去。

【例】Down it flew.

它“刷”地飞了下来。

（12）为了保持句子平衡、突出句子的状语或使句子之间联系得更紧密，也可使用倒装语序。请看下例：

【例】Such would be our home in the future.

这就是我们未来的家庭。

【例】On both sides of the street are tall trees.

街道两旁树木林立。

【例】Lying on the ground was a boy of sixteen.

躺在地上的是一个十六岁的男孩。

【例】Very important to him is to learn how to repair printers.

对他来讲，最重要的是学会如何修理打印机。

Exercises

一、根据课文内容，判断以下叙述的正误。

(1) The Java programming language is a low-level language.

(2) The Java programming language is unusual because a program is both compiled and interpreted.

(3) Every Java interpreter is an implementation of the Java VM.

(4) A Java interpreter can be a development tool or a Web browser.

(5) Native code is a bit slower than the Java platform can be a bit slower than.

(6) The Java platform is a platform-independent environment.

(7) A server serves and supports clients on a network.

(8) A servlet can almost be thought of as an applet that runs on the browser side.

(9) Servlets are similar to applets in that they are runtime extensions of applications.

(10) Servlets run within proxy servers, configuring or tailoring the server.

二、根据课文内容填空。

(1) Java technology is ____.

(2) A platform is ____.

(3) The Java platform is ____.

(4) The Java platform has two components: ________________ and ________________.

(5) The most common types of programs written in the Java programming language are __________ and ________________.

(6) You can write many types of programs by using ________________.

(7) An application is ____________________________________.

(8) Examples of servers are __________________________________.

三、把下列句子改成倒装句。

(1) If the design had been ready, we should have started working yesterday.

(2) Though the problem is complicated, a computer can work it out in a few minutes.

(3) The manager didn't leave his office until it was dark.

(4) The horse went off.

(5) Without steel there would be no trains, no ships, no cars or planes. There would be no electrical motors, telephones or televisions, either.
(6) We can't consider ordinary air as a good conductor in any way.
(7) You only learn a foreign language well if you practice all the four skills of listening, speaking, reading and writing.
(8) I had hardly arrived when I had a new problem to cope with.
(9) She had seldom met with this kind of trouble.
(10) A beautiful young girl lay on the bed.

四、从供选择的词汇中选择最合适的填在文中相应数字处。

Programming is a ___1___ that has various engineering and ___2___ aspects. Although most of aspects are interrelated, each of them covers number of ___3___ and goals which are very specific. The progress being made in teaching ___4___ is reflected by the fact that most ___5___ science curricula contain a broad spectrum of programming courses, each emphasizing a different aspect of programming.

供选择的答案：

A. notation
B. physical
C. function
D. discipline
E. philosophy
F. software engineering
G. characteristics
H. mathematical
I. programming
J. program
K. method
L. computer

五、听短文，在画线处填写所听到的单词或词组。

___1___ is a programming language originally developed by Sun Microsystems and released in 1995 as a ___2___ component of Sun Microsystems'Java platform. The language derives much of its ___3___ from C and C++ but has a simpler ___4___ model and fewer low-level facilities. Java applications are typically compiled to ___5___ that can run on any Java ___6___ machine (JVM) regardless of computer architecture.

The original and reference implementation Java compilers, virtual machines, and ___7___ libraries were developed by Sun from 1995. As of May 2007, in compliance with the ___8___ of the Java Community Process, Sun made available most of their Java technologies as ___9___ software under the GNU General Public License. Others have also developed alternative ___10___

of these Sun technologies, such as the GNU Compiler for Java and GNU Classpath.

六、计算机软件水平考试真题自测（高级程序员级）：选择填空。

The Rational Unified Process (RUP) is a software engineering process, which captures many of best practices in modern software development. The notions of ___1___ and scenarios have been proven to be an excellent way to capture function requirements. RUP can be described in two dimensions-time and content. In the time dimension, the software lifecycle is broken into cycles. Each cycle is divided into four consecutive ___2___ which is concluded with a well-defined ___3___ and can be further broken down into ___4___ a complete development loop resulting in a release of an executable product, a subset of the final product under development, which grows incrementally to become the final system. The content structure refers to the disciplines, which group ___5___ logically by nature.

1. A. artifacts	B. use-cases	C. actors	D. workers
2. A. orientations	B. views	C. aspects	D. phases
3. A. milestone	B. end-mark	C. measure	D. criteria
4. A. rounds	B. loops	C. iterations	D. circularities
5. A. functions	B. workflows	C. actions	D. Activities

Skill Training

计算机英文论文摘要概述

文章摘要是对所写文章主要内容的精炼概括。美国人称摘要为“Abstract”，而英国人则喜欢称其为“Summary”。

写摘要时，应用最为简练的语言来表达论文之精华。论文摘要的重点应放在所研究的成果和结论上。摘要应尽量简练，多采用第三人称。

1. 摘要的种类（Type of Abstracts）

摘要分为信息性摘要与叙述性摘要，或者两者的结合。

1.1 信息性摘要（Informative Abstracts）

信息性摘要包括原始文献某些重要内容的梗概，它是论文全文的浓缩，可以替代阅读全文。

信息性摘要主要由以下三部分组成。

（1）目的：主要说明作者写此文章的目的，或说明本文解决的问题。

（2）过程及方法：说明主要工作过程及所用的方法，也包括众多的边界条件，使用的主要设备和仪器等。

（3）结果和结论：作者在此工作过程最后得到的结果和结论，应体现创新，突出特殊贡献。如有可能，尽量提一句作者所得结果和结论的应用范围和应用情况。

有些杂志明确提出“结构化摘要”，包括：Background，Methods，Results，Conclusion。

信息性摘要多用于科技杂志或科技期刊的文章，也用于会议录中的会议论文及各种专题技术报告。

1.2　叙述性摘要（Descriptive Abstracts）

叙述性摘要仅指出文献的综合内容，适用于综述性文献。综述性文献最常见的如某技术在某时期的综合发展情况或某技术在目前的发展水平及未来展望等。总之，这种文献是综述情况而不是针对某个具体技术工艺、产品或设备。

一般情况下，信息性摘要占有绝大多数。

2. 摘要长度（Length of the Abstracts）

摘要长度一般不超过 150 个单词，不少于 100 个单词。特殊情况下可以例外，视原文文献而定。

一般缩短摘要方法如下：

（1）取消不必要的字句，例如：It is reported…；Extensive investigations show that…。

（2）取消或减少背景情况（Background Information）。

（3）只表示新情况和新内容，取消过去的研究细节。

（4）不说废话和空话，例如，“本文所谈的有关研究工作是对过去老编程方法的一个极大的改进”不可进入摘要。

（5）作者在文献中谈及的未来计划不要纳入摘要；第一句话切不可重复题目（Title）。

3. 英文摘要的时态与语态

说明研究目的，通常用一般现在时或一般过去时。说明方法和结果，一般用一般过去时、第三人称的被动语态表达。结论通常用一般现在时、被动语态表达，也可用主动语态表达。

4. 英文摘要常用句型

4.1　表达论文内容的常用句型

4.1.1　第三人称主动语态

This paper (article) describes (reports, presents, discusses)	……本文叙述（报告，介绍，讨论）……
The author discusses…	作者讨论……

常用的动词还有 analyze（分析），evaluate（评价），compare（比较），describe（描述）等。

4.1.2　第一人称主动语态

We report (on) …	我们报告……
We describe a case of…	我们描述一例……
In this paper, we present…	本文介绍……

4.1.3　一般现在时被动语态

A case is reported in which…	本文报告一例……
A study of…is reported	本文报告……的研究

…is（are）described. 本文描述……

In this paper，… is（are）presented. 本文介绍……

4.2 表达目的常用句型

4.2.1 一般过去时、被动语态

The purpose（aim，objective）of this study was to… 本文研究旨在……

The goal of this investigation was to… 本文研究旨在……

This study was designed（undertaken）to… 本文研究旨在……

This prospective study was performed to… 本文前瞻性研究的目的是……

A study to…was carried out（during…） （在……期间）所作研究的目的是……

An attempt has been made to… 为了……而做试验

4.2.2 动词不定式短语

To evaluate，report，investigate，study，analyze…

4.2.3 一般过去时、主动语态

We（the authors）conducted a study to… 为了……我们进行了研究

To determine…，we studied… 为了确定……，我们研究了……

In an attempt to…，in an effort to…或 in order to…，we carried out a pilot study… 为了……，我们进行了……的初步研究

表示目的的常用动词有：evaluate（评价），examine（检查，观察），determine（确定，查明），elucidate（阐明），explore（探索），test（测试），compare（比较），estimate（评估），assess（估价），investigate（调查）等。

4.3 表达方法的常用句型

4.3.1 方法

表达研究类型 prospective，retrospective，cohort，case-control，in vivo，in vitro。

Using…（technique），we studied… 我们用……（技术）研究了……

Using…，it was found that… （我们）用……发现了……

…was（were）measured using… （我们）用……测定了……

…was（were）analyzed（reviewed）by… （我们）用……分析（回顾）了……

…was（were）treated with… （我们）用……治疗了……

…measurements were made of… 测定了……

常用的动词还有：study，measure，determine，investigate，isolate，demonstrate，examine，identify 等。

4.3.2 分组（主语通常为 patients，subjects，animals）

…were randomly divided（grouped）into…groups ……被随机分成……组

…were separated into…groups based on… 根据……，将……分成……组

The groups were as follows： 分组如下：

4.4 表达结果的常用句型

4.4.1 表达结果

The results showed（demonstrated）that… 结果表明……

It was found that… （我们）发现……

It was observed that… （我们）观察到……

第 2 和 3 句也可改为第一人称：We found that…和 We observed that…。

4.4.2 表达增加或减少

…decreased by（40%）… 降低（40%）……

…（a 70%）reduction in…was observed ……观察到……降低（70%）

There was a（15%）elevation in… ……增高（15%）

…resulted in（a marked）increase in… ……导致……（明显）增高

…was lowered from…to… ……从……下降到……

表达增加或减少的常用动词还有 rise，raise，decline，drop，fall，lower 等，可用 by 表示是净增/减的数或倍数。例如“…increase by 40%”，表示“增加 40%”，而 to 表示增加或减少到某个具体数值。

4.4.3 表达相关与差别

There was a significant linear correlation between…and… ……与……有显著的线性相关性

…showed no strong correlation between…and… ……显示……与……无密切相关性

…correlated positively with… ……与……呈正相关

A negative correlation was found between…and… 发现……与……呈负相关

There was（were）no significant difference（s）or（No）significant difference（s）was found（observed）between…and…

……与……之间……（无）有显著差别

…was closely related to… ……与……密切相关

表示相互关系的常用词还有 relationship（关系），association（联系），regression（回归）。

4.4.4 表达改善或保持

…demonstrated a significant improvement in… ……方面呈明显改善

…values returned to levels ……值恢复到水平

…remained at a mean of… ……保持在……平均值水平

4.5 表达结论的常用句型

4.5.1 常用提示句型

These results suggest that… 结果提示……

These findings indicate that… 这些发现表明……

The data（study）show the need for… 数据（研究）表明需要……

The results support the concept that…	结果支持……的概念
Our observations confirm that…	我们的观察证实……

表示提示的常用词还有 imply（提示），demonstrate（证明）和 illustrate（说明）。

4.5.2 做出结论或建议

We conclude that…	我们的结论是……
We suggest (believe, postulate) that…	我们建议（认为，设想）……
It is concluded that…	结论是……
It is suggested (proposed, recommended,) that…	提示（建议）……（虚拟语气）
It is estimated that…	估计……

4.5.3 表示一致性

These results accord with…	结果与……一致
The results are concordant with…	结果与……一致
The results agree well with…	结果与……一致
be consistent with, in accordance with…	与……一致
to parallel	与……相平行

4.5.4 结果转向结论

Thus, therefore, hence…	……因此，……
In conclusion…	最后，……
From this study, we conclude that…	根据本项研究，我们的结论是……
These findings, coupled with the observation that…, suggest that…	这些发现连同对……的观察结果一起，提示……
In the light of this experience we therefore conclude that…	根据这个经验，我们得出……的结论

4.5.5 表达今后方向

The mechanisms by which…remain to be investigated	关于……的机制仍需研究
Further studies are necessary to (evaluate) …	有必要进一步研究，以（评价）……

Reading Material

Object-Oriented Analysis, Design and Object-Oriented Modeling

1. Object-oriented analysis and design

Object-oriented analysis and design (OOAD)[1] is a software engineering approach that models a system as a group of[2] interacting objects. Each object represents some entity of interest in the

[1] OOAD (Object-oriented analysis and design) 面向对象分析与设计

[2] a group of 一群

system being modeled, and is characterised by its class, its state (data elements), and its behavior. Various[1] models can be created to show the static structure[2], dynamic behavior, and run-time deployment of these collaborating objects. There are a number of different notations for representing these models, such as the Unified Modelng Language (UML).

Object-oriented analysis (OOA) applies object-modeling techniques to analyze the functional requirements for a system. Object-oriented design (OOD) elaborates[3] the analysis models to produce implementation specifications. OOA focuses on what the system does, OOD on how the system does it.

1.1 Object-oriented systems

An object-oriented system is composed[4] of objects. The behavior of the system results from the collaboration of those objects. Collaboration between objects involves them sending messages to each other. Sending a message differs from calling a function in that when a target[5] object receives a message, it itself decides[6] what function to carry out[7] to service that message. The same message may be implemented by many different functions, the one selected depending on the state of the target object.

The implementation of "message sending" varies depending on the architecture of the system being modeled, and the location of the objects being communicated with.

1.2 Object-oriented analysis

Object-oriented analysis (OOA) looks at the problem domain, with the aim[8] of producing a conceptual[9] model of the information that exists in the area being analyzed. Analysis models do not consider any implementation constraints that might exist, such as concurrency[10], distribution, persistence, or how the system is to be built. Implementation constraints are dealt with during object-oriented design (OOD). Analysis is done before the design.

The sources for the analysis can be a written requirements statement, a formal[11] vision document, interviews with stakeholders or other interested parties. A system may be divided into multiple domains, representing different business, technological, or other areas of interest, each of which are analyzed separately.

The result of object-oriented analysis is a description of what the system is functionally required to do, in the form of a conceptual model. That will typically be presented as a set of use cases, one

---

[1] various [ˈvɛəriəs] *adj.* 不同的，各种各样的
[2] static structure 静态结构
[3] elaborate [iˈlæbərəit] *v.* 详细描述
adj. 精心制作的，详细阐述的，精细
[4] compose [kəmˈpəuz] *v.* 组成
[5] target [ˈtɑːgit] *n.* 目标，对象
[6] decide [diˈsaid] *v.* 决定
[7] carry out 完成，实现，执行
[8] aim [eim] *n.* 目标，目的，瞄准
[9] conceptual [kənˈseptʃuəl] *adj.* 概念上的
[10] concurrence [kənˈkʌrənsi] *n.* 并发性
[11] formal [ˈfɔːməl] *adj.* 正式的，形式的

or more UML class diagrams, and a number of interaction diagrams. It may also include some kind of user interface mock-up[1].

1.3 Object-oriented design

Object-oriented design (OOD) transforms the conceptual model produced in object-oriented analysis to take account of the constraints imposed by the chosen architecture and any non-functional — technological or environmental-constraints, such as transaction throughput[2], response time[3], run-time platform[4], development environment, or programming language.

The concepts in the analysis model are mapped[5] onto implementation classes and interfaces. The result is a model of the solution domain, a detailed description of how the system is to be built.

2. Object-oriented modeling

Object-oriented modeling, or OOM[6], is a modeling paradigm mainly used in computer programming. Prior to the rise of OOM, the dominant paradigm was functional programming, which emphasized[7] the use of discreet reusable[8] code blocks[9] that could stand on their own, take variables, perform a function on them, and return values.

The Object-oriented paradigm assists the programmer to address the complexity of a problem domain by considering the problem not as a set of functions that can be performed but primarily as a set of related, interacting objects. The modeling task then is specifying, for a specific context, those objects (or the class the objects belongs to), their respective[10] set of properties and methods, shared by all objects members of the class.

As an example, in a model of a Payroll System, a company is an object. An employee is another object. Employment is a relationship or association. An employee class (or object for simplicity) has attributes like name, birthdate, etc. The association itself may be considered as an object, having attributes, or qualifiers like position, etc. An employee method may be promotion, raise, etc.

The model description or schema[11] may grow in complexity to require a notation. Many notations have been proposed, based on different paradigms, diverged[12], and converged in a more popular one known as UML.

An informal description or a schema notation is translated by the programmer or a computer-

---

[1] mock-up [ˈmɔkʌp] *n.* 实验或教学用的实物大模型
[2] throughput [ˈθruːput] *n.* 生产量，生产能力，吞吐量
[3] response time 响应时间
[4] run-time platform 实时平台
[5] map [mæp] *n.* 映射
[6] Object-oriented modeling (OOM) 面向对象建模
[7] emphasize [ˈemfəsaiz] *vt.* 强调，着重
[8] reusable [riːˈjuːzəbl] *adj.* 可重用的，可复用的
[9] code block 代码段，程序段
[10] respective [risˈpektiv] *adj.* 分别的，各自的
[11] schema [ˈskiːmə] *n.* 大纲或模型；图解，图表
[12] diverge [daiˈvəːdʒ] *vi.* （道路等）分叉，（意见等）分歧

aided software engineering tool in the case of schema notation (created using a module specific to the CASE[1] tool application) into a specific programming language that supports object-oriented programming (or a class type), a declarative language or into a database schema.

参 考 译 文

关于 Java 技术

Java 技术既是一种编程语言，也是一个平台。

1. Java 编程语言

Java 编程语言是一种高级语言，其特点可以用以下术语来描述：

- 简单；
- 体系中立；
- 面向对象；
- 可移植；
- 分布式的；
- 高性能；
- 解释的；
- 多线程；
- 健壮的；
- 动态的；
- 安全的。

上述每一个术语在 James Gosling 和 Henry McGilton 写的白皮书《Java 语言环境》中都有解释。

应用大多数编程的语言，可以编译也可以解释一个程序，以便在计算机上运行它。Java 编程语言与众不同之处在于程序既可以被编译也可以被解释。当编译时，首先把程序转换为一个称为“Java 字节码”的中间语言，这个平台独立的代码可以被 Java 平台上的任何解释程序来解释。该解释程序解析和运行计算机上的每个 Java 字节代码。只编译一次。每当运行程序时就解释。下图表明它是如何工作的：

可以把 Java 字节代码看作用于 Java 虚拟机（Java VM）的机器代码指令。每个 Java 解释程序，无论是开发工具还是能运行 applet 的 Web 浏览器，都可以被 Java 虚拟机执行。

Java 字节代码帮助实现了“一次写成，到处可用”的可能。可以在带有 Java 编译器的任何平台上把程序编译为字节代码。然后字节代码可以在任何 Java 虚拟机上运行。这意味着只要带 Java 虚拟机，用 Java 编程语言编写的同一个程序可以运行在 Windows2000、Solaris 工作站或 iMac 上。

（图略）

---

[1] CASE（computer-aided software engineering）计算机辅助软件工程

2. Java 平台

平台是程序运行的硬件或软件环境。我们已经提及一些最流行的平台，如 Windows2000、Linux、Solaris 和 MacOS。大多数平台都可以描述为操作系统和硬件的组合。

Java 平台不同于其他大部分平台，它是一个运行在基于其他硬件平台上的纯软件平台。Java 平台由两部分组成：

- Java 虚拟机（Java VM）；
- Java 应用编程接口（Java API）。

已经了解了 Java 虚拟机。它是 Java 平台的基础并给其他基于硬件的平台提供端口。

Java API 是制成软件部件的一个大集。这些部件提供了许多有用的性能，如图形用户接口部件。Java API 被组成为与类和接口相关的库，这些库称为“包”。

下图描述了在 Java 平台上运行的程序。如图所示，Java API 和虚拟机把程序与硬件隔离了。

（图略）

本地代码是编译后的代码，编译了的代码可以运行在特定的硬件平台上。作为平台独立的环境，Java 平台可能比本地代码运行得略慢一些。然而，智能编译器、协调良好的解释程序和即时代码编译器的性能可以接近本地代码，而不损害可移植性。

3. Java 技术可以做什么?

用 Java 编程语言编写的大多数程序都是 applet 和应用程序。如果你在 Web 上冲过浪，或许你已经熟悉了 applet。applet 是符合一定约定的程序，这些约定使它可以运行于可用 Java 的浏览器内。在片头显示用 Java 技术制作的向你招手的吉祥物（Duke）动画。

然而，Java 编程语言不仅仅用于编写 Web 上娇小可爱的娱乐性小应用程序。通用的、高级 Java 编程语言也是一个功能强大的软件平台。使用丰富的 API，可以编写多种程序。

一个应用程序是可以直接运行于 Java 平台的独立程序的。一种特殊的应用程序称为“server”，它对网络客户提供服务和支持。例如：Web 服务器、代理服务器、邮件服务器和打印服务器。另一特殊的程序称为“servlet”，它几乎可以被看作是运行在服务器端的 applet。Java Servlet 是建立交互网络应用的流行选择，用来替代 CGI 脚本。Servlet 类似于 applet，在应用程序运行时扩展。虽然它不是工作在浏览器内，但运行在 JavaWeb 服务器内，配置或定制服务器。

API 是怎样支持各种程序的呢？它用一个提供各种功能的软件部件包来做到这一点的。每个 Java 平台都可以完整地提供如下功能。

- 基本要素：对象、串、线程、数字、输入和输出、数据结构、系统性能、日期和时间等。
- Applets：applet 所用的惯例集。
- 网络：URL（统一资源定位符）、TCP（传输控制协议）、UDP（用户数据报协议）套接字及 IP（网际协议）地址。
- 国际化：使所写的程序可以让世界各地的用户使用。程序可以自动适应特定的场所并以适当的语言显示。

- 安全：有低级和高级两种，包括电子签名、公共和私有钥匙管理、访问控制和发证书。
- 软件部件：也称为 JavaBeans™，可以插入到现有部件的体系结构中。
- 对象连续化：允许轻量持续并通过远程方法调用来通信。
- Java 数据库连接（JDBC™）：对各种相关数据库提供统一的访问。

Java 平台也有可以用于二维和三维图形、可访问性、服务器、协作、电话技术、语音、动画等的 API。下图描述了 Java2SDK 所包含的内容。

（图略）

注意：Java2SDK 标准版本 1.3 版。Java2 的运行环境（JRE）由以下几部分组成：虚拟机、Java 平台核心类及支持文件。Java2SDK 包括 JRE 和像编译器及调试器这样的开发工具。

Lesson 8

Text

Basic Concepts of Database

1. Database

A database is a collection of information that is organized so that it can easily be accessed, managed and updated. Databases can be classified according to types of content: bibliographic, full-text, numeric and images.

In computing, databases are sometimes classified according to their organizational approach. The most prevalent approach is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. A distributed database is one that can be dispersed or replicated among different points in a network. An object-oriented programming database is one that is congruent with the data defined in object classes and subclasses.

Computer databases typically contain aggregations of data records or files, such as sales transactions, product catalogs and inventories, and customer profiles. Typically, a database manager provides users the capabilities of controlling read/write access, specifying report generation and analyzing usage. Databases and database managers are prevalent in large mainframe systems, but are also present in smaller distributed workstation and mid-range systems such as the AS/400 and on personal computers.

2. Relational Database

A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables. The relational database was invented by E. F. Codd at IBM (International Business Machines Corporation) in 1970.

The standard user and application program interface to a relational database is the structured query language (SQL). SQL statements are used both for interactive queries for information from a relational database and for gathering data for reports.

In addition to being relatively easy to create and access, a relational database has the important advantage of being easy to extend. After the original database creation, a new data category can be added without requiring that all existing applications be modified.

A relational database is a set of tables containing data fitted into predefined categories. Each

table (which is sometimes called a relation) contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns. For example, a typical business order entry database would include a table that described a customer with columns for name, address, phone number, and so forth. Another table would describe an order: product, customer, date, sales price, and so forth. A user of the database could obtain a view of the database that fitted the user's needs. For example, a branch office manager might like a view or report on all customers that had bought products after a certain date. A financial services manager in the same company could, from the same tables, obtain a report on accounts that needed to be paid.

When creating a relational database, you can define the domain of possible values in a data column and further constraints that may apply to that data value. For example, a domain of possible customers could allow up to ten possible customer names but be constrained in one table to allowing only three of these customer names to be specifiable.

The definition of a relational database results in a table of metadata or formal descriptions of the tables, columns, domains and constraints.

3. SQL

SQL is a standard language for making interactive queries from a database and updating a database such as IBM's DB2, Microsoft's Access and database products from Oracle, Sybase and Computer Associates. Although SQL is both an ANSI (American National Standards Institute) and an ISO (International Organization for Standardization) standard, many database products support SQL with proprietary extensions to the standard language. Queries take the form of a command language that lets you select, insert, update, find out the location of data and so forth. There is also a programming interface.

4. Database Management System

A database management system (DBMS), sometimes just called a database manager, is a program that lets one or more computer users create and access data in a database. The DBMS manages user requests (and requests from other programs) so that users and other programs are free from having to understand where the data is physically located on storage media and, in a multi-user system, which else may also be accessing the data. In handling user requests, the DBMS ensures the integrity of the data (that is, making sure it continues to be accessible and is consistently organized as intended) and security (making sure only those with access privileges can access the data). The most typical DBMS is a relational database management system (RDBMS). A standard user and program interface is the SQL. A newer kind of DBMS is the object-oriented database management system (ODBMS).

A DBMS can be thought of as a file manager that manages data in databases rather than files in file systems. In IBM's mainframe operating systems, the non-relational data managers were (and are, because these legacy application systems are still used) known as access methods.

A DBMS is usually an inherent part of a database product. On PCs, Microsoft's Access is a

popular example of a single or small-group user DBMS. Microsoft's SQL Server is an example of a DBMS that serves database requests from multiple (client) users. Other popular DBMSs (these are all RDBMSs, by the way) are IBM's DB2, Oracle's line of database management products, and Sybase's products.

IBM's Information Management System (IMS) was one of the first DBMSs. A DBMS may be used by or combined with transaction managers, such as IBM's Customer Information Control System (CICS).

5. Distributed Database

A distributed database is a database in which portions of the database are stored on multiple computers within a network. Users have access to the portion of the database at their location so that they can access the data relevant to their tasks without interfering with the work of others.

6. DDBMS

A DDBMS (distributed database management system) is a centralized application that manages a distributed database as if it were all stored on the same computer. The DDBMS synchronizes all the data periodically, and in cases where multiple users must access the same data, ensures that updates and deletes performed on the data at one location will be automatically reflected in the data stored elsewhere.

7. Field

In a database table, a field is a data structure for a single piece of data. Fields are organized into records, which contain all the information within the table relevant to a specific entity. For example, in a table called customer contact information, telephone number would likely be a field in a row that would also contain other fields such as street address and city. The records make up the table rows and the fields make up the columns.

8. Record

In a database, a record (sometimes called a row) is a group of fields within a table that are relevant to a specific entity. For example, in a table called customer contact information, a row would likely contain fields such as: ID number, name, street address, city, telephone number and so on.

9. Table

In a relational database, a table (sometimes called a file) organizes the information about a single topic into rows and columns. For example, a database for a business would typically contain a table for customer information, which would store customers' account numbers, addresses, phone numbers, and so on as a series of columns. Each single piece of data (such as the account number) is a field in the table. A column consists of all the entries in a single field, such as the telephone

numbers of all the customers. Fields, in turn, are organized as records, which are complete sets of information (such as the set of information about a particular customer), each of which comprises a row. The process of normalization determines how data will be most effectively organized into tables.

New Words

database	[ˈdeitəbeis]	*n.* 数据库
collection	[kəˈlekʃən]	*n.* 集合
organize	[ˈɔːɡənaiz]	*v.* 组织
access	[ˈækses]	*n.* 访问
		vt. 存取
classify	[ˈklæsifai]	*vt.* 分类，分等
bibliographer	[ˌbibliˈɔɡrəfə]	*adj.* 目录的
approach	[əˈprəutʃ]	*n.* 方法，步骤，途径，通路
tabular	[ˈtæbjulə]	*adj.* 制成表的，扁平的，表格式的，平坦的
		vi. 列表，排成表格式
distributed	[disˈtribjuːtid]	*adj.* 分布式的
disperse	[disˈpəːs]	*v.* （使）分散，（使）散开
replicate	[ˈreplikit]	*v.* 复制
congruent	[ˈkɔŋgruənt]	*adj.* （与 with 连用）一致的，适合的
aggregation	[æɡriˈɡeiʃən]	*n.* 集合，集合体，聚合
catalog	[ˈkætəlɔɡ]	*n.* 目录，目录册
		v. 编目录
capability	[ˌkeipəˈbiliti]	*n.* （实际）能力，性能，容量
analyze	[ˈænəlaiz]	*vt.* 分析，分解
prevalent	[ˈprevələnt]	*adj.* 普遍的，流行的
set	[set]	*n.* 集合，集
reorganize	[ˈriːˈɔːɡənaiz]	*v.* 改组，再编制，改造
query	[ˈkwiəri]	*v.* 询问，查询
extend	[iksˈtend]	*v.* 扩充，延伸，伸展
column	[ˈkɔləm]	*n.* 列，纵队
row	[rau]	*n.* 行，排
predefine	[ˈpriːdiˈfain]	*vt.* 预先确定，预定义
instance	[ˈinstəns]	*n.* 实例，例证
view	[vjuː]	*n.* 视图
domain	[dəuˈmein]	*n.* 域，范围
constraint	[kənˈstreint]	*n.* 约束，限制
specifiable	[ˈspesifaiəbl]	*adj.* 能指定的；能详细说明的；能列举的

metadata	[ˈmetəˈdeitə]	*n.* 元数据
Oracle	[ˈɔrəkl]	*n.* 美国甲骨文公司，主要生产数据库产品
command	[kəˈmɑːnd]	*n.&v.* 命令
insert	[inˈsəːt]	*vt.* 插入
update	[ʌpˈdeit]	*v.* 更新
ensure	[inˈʃuə]	*v.* 确保
privilege	[ˈprivilidʒ]	*n.* 特权
inherent	[inˈhiərənt]	*adj.* 固有的，内在的
client	[ˈklaiənt]	*n.* 顾客，客户，委托人
centralize	[ˈsentrəlaiz]	*vt.* 集聚，集中
synchronize	[ˈsiŋkrənaiz]	*v.* 同步
periodically	[ˌpiəriˈɔdikəli]	*adv.* 周期性地，定时性地
delete	[diˈliːt]	*vt.* 删除
automatically	[ɔːtəˈmætikli]	*adv.* 自动地
reflect	[riˈflekt]	*v.* 反射，反映，表现
field	[fiːld]	*n.* 域
topic	[ˈtɔpik]	*n.* 主题，题目
series	[ˈsiəriːz]	*n.* 连续，系列
complete	[kəmˈpliːt]	*adj.* 完备的，完全的，完成的
normalization	[ˌnɔːməlaiˈzeiʃən]	*n.* 正常化，标准化

Phrases

tabular database	表格数据库
a number of	许多的
distributed database	分布式数据库
customer profile	客户简介
find out	找出；发现
account number	账号
in turn	依次，轮流

Abbreviations

SQL (structured query language)	结构化查询语言
IBM (International Business Machines Corporation)	国际商用机器公司
ANSI (American National Standards Institute)	美国国家标准协会
ISO (International Organization for Standardization)	国际标准化组织
DBMS (Database Management System)	数据库管理系统

RDBMS（Relational Database Management System）	关系型数据库管理系统
ODBMS（object-oriented database management system）	面向对象的数据库管理系统
IMS（Information Management System）	信息管理系统
CICS（Customer Information Control System）	客户信息管理系统
DDBMS（distributed database management system）	分布式数据库管理系统

Notes

[1] A relational database is a collection of data items organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables.

本句中，organized as a set of formally-described tables 是一个过去分词短语作定语，修饰和限定 data items，它可以扩展成一个定语从句：which are organized as a set of formally-described tables。from which data can be accessed or reassembled in many different ways without having to reorganize the database tables 是一个介词前置的定语从句，修饰和限定 formally-described tables。

[2] A database management system（DBMS），sometimes just called a database manager，is a program that lets one or more computer users create and access data in a database.

本句中，sometimes just called a database manager 对 a database management system（DBMS）作进一步补充说明。that lets one or more computer users create and access data in a database 是一个定语从句，修饰和限定 a program。在该从句中，create and access data in a database 是一个不带 to 的动词不定式短语作宾语 one or more computer users 的补足语。

英语中，在 make、let、have、see、hear、watch、notice、feel 等动词后面用动词不定式作宾语补足语时，不定式都不带 to。但当宾语补足语变成主语补足语时，to 则不能省略。请看下例：

His boss often makes him work on weekends without extra pay.

他老板经常让他周末加班，却不给他额外报酬。

Let each man decide for himself.

让每个人自己决定。

Someone was heard to come up the stairs.

听见有人上楼。

[3] A DDBMS（distributed database management system）is a centralized application that manages a distributed database as if it were all stored on the same computer.

本句中，that manages a distributed database as if it were all stored on the same computer 是一个定语从句，修饰和限定 a centralized application。在该从句中，as if it were all stored on the same computer 是一个方式状语从句。

英语中，as if 和 as though 引导的方式状语从句一般要用虚拟语气。请看下例：

He talks as if he were a knowing-all.

他说起话来好像是一个百事通。

[4] The DDBMS synchronizes all the data periodically，and in cases where multiple users must access the same data，ensures that updates and deletes performed on the data at one location will be

automatically reflected in the data stored elsewhere.

本句中，in cases where multiple users must access the same data 作条件状语。that updates and deletes performed on the data at one location will be automatically reflected in the data stored elsewhere 是一个宾语从句，作 ensures 的宾语。在该从句中，performed on the data at one location 是一个过去分词短语作定语，修饰和限定 updates and deletes。

[5] For example, in a table called customer contact information, telephone number would likely be a field in a row that would also contain other fields such as street address and city.

本句中，called customer contact information 是一个过去分词短语，作定语，修饰和限定 a table。that would also contain other fields such as street address and city 是一个定语从句，修饰和限定 a row。

Grammar

It 的 用 法

it 可用作代词，以代替除人以外的一切生物及无生命的东西；也可用作引导词，在句子中作形式主语或形式宾语。it 还可以构成强调句型。

1. it 作代词

1）作人称代词

it 作人称代词时用于除人之外的一切生物及无生命的东西。请看下例：

【例】This is a computer. We made it ourselves. It works well. Its performance is good.

这是一台计算机。是我们自己制造的。它很好用。其性能优良。

【例】The sun is our great source of energy. Without it there would be no life on the earth.

太阳是我们的巨大能源。没有它地球上就不会有生命。

it 有时也可用于指人。请看下例：

【例】Who is that? It's me.

谁呀？是我。

2）作指示代词

it 表示“这”、“那”。请看下例：

【例】What is this? It's a laser printer.

这是什么？这是一台激光打印机。

【例】What's that? It's a P4 computer.

那是什么？那是一台 P4 计算机。

3）作非人称代词

it 表示时间、距离、量度和自然现象等。

(1）表示时间。请看下例：

【例】It's ten o'clock now.

现在十点了。

【例】It was nearly two in the morning before she fell asleep.

快凌晨两点她才睡着。

【例】It's two years now since he came to this company.

他来到这家公司已经两年了。

【例】It's high time we went.

我们该走了。

(2）表示距离。请看下例：

【例】It's about 150 million kilometers from the earth to the sun.

地球到太阳的距离约一亿五千万公里。

【例】It's a very long way to the Great Wall.

到长城还远着呢。

(3）表示量度。请看下例：

【例】It's 220 volts in voltage.

电压为 220 伏。

(4）表示天气和温度。请看下例：

【例】It's still raining.

天还在下雨。

【例】It's very cold today.

今天天气非常冷。

【例】It was raining cats and dogs when we got home.

我们到家时，天正下着倾盆大雨。

【例】It's a lovely day, isn't it?

今天天气真好，不是吗？

【例】It's 36 degrees Celsius. Wow! That's hot.

今天 36 摄氏度。噢！真热。

(5）表示前面已经提到的或将会发生的某件事情。请看下例：

【例】——Do you enjoy playing football?

——No, I hate it.

——你喜欢踢足球吗？

——不，我讨厌足球。

句中，it 代替前面已经提到的 playing football。

【例】Tom solved the problem, and it made him happy.

汤姆解决了那个问题，这使他很高兴。

句中，it 指 Tom solved the problem 这件事。

【例】She knew that she would hate it if they said no.

她知道她不希望他们不同意。

句中，it 表示将要发生的事。

2. 作引导词

1）作形式主语

当句子的主语是动词不定式、动名词或主语从句时，通常把它放在谓语之后，而将 it 放在句首作形式主语。请看下例：

【例】It's nice to be with you.

和你在一起我觉得很好。

句中，it 是形式主语，而真正的主语是 to be with you。

【例】It's not easy to make our manager change his mind.

要我们经理改变主意是不容易的。

句中，真正的主语是动词不定式 to make our manager change his mind。

【例】It's no use trying to explain. He is not interested.

解释也没有用。他不感兴趣。

句中，真正的主语是动名词短语 trying to explain。

【例】It's nice lying in the sun on such a warm day.

这么暖和的日子里躺在阳光下很舒服。

句中，真正的主语是 lying in the sun on such a warm day。

【例】It's a pity that Peter didn't come.

彼得没有来，真遗憾。

句中，真正的主语是 that Peter didn't come。

【例】Is it true that she has finished her design?

她的设计已经完成，这是真的吗？

【例】It doesn't matter when you go.

你什么时候走无关紧要。

句中，真正的主语是 when you go。

【例】It's very clear what he meant.

他的意思十分清楚。

【例】It occurred to me that he might have repaired the plotter.

我忽然想到他可能已经修好了绘图机。

【例】It seems obvious that you must finish this project as soon as possible.

显而易见，你必须尽快完成这项工程。

2）作形式宾语

当句子的宾语是动词不定式、动名词或宾语从句且句子的宾语还要有一个补足语时，通常把 it 放在动词之后，作形式宾语，而把真正的宾语放在宾语补足语之后。请看下例：

【例】I found it difficult to translate this sentence into English.

我发现要把这个句子译成英语很难。

【例】She thought it easy to learn to type.

她认为学习打字很容易。

【例】I think it useless waiting. He won't come.

我认为等是没有用的。他不会来了。

【例】He thought it a pity that I didn't go to the exhibition yesterday.

他认为我昨天没有去参加展览会是一件很遗憾的事。

【例】He made it clear that he disagreed to this plan.

他明确地表示他不同意这项计划。

3. 构成强调句型

构成 It is (was) … that (who) 强调句型，着重强调句子的某一部分，如主语、宾语或状语。

1) 强调主语

【例】It was Mr. Smith that solved this problem.

是史密斯先生解决了这个问题。

【例】Is it carbon that makes steel so strong and hard?

是碳使钢变得如此坚硬的吗?

【例】It was she that helped us a lot.

是她给了我们很大的帮助。

句中，she 不能换成 her。

【例】It was we who arrived there first.

是我们先到那儿的。

句中，we 不能换成 us。

这里值得注意的是，当强调的主语是人时，强调句型中的 that 可以换成 who。当主语是人称代词时，一定要注意人称代词的格：在正式的英语中，只能用主格；而在非正式的英语中，有时也可使用宾格。换言之，在如 TOEFL、EPT、GRE 等考试中，只能使用主格。

2) 强调宾语

【例】It is light and heat that the sun gives us.

太阳供给我们光和热。

【例】It was three printers that they bought last week.

他们上周买了三台打印机。

注意，句中的 was 不能换成 were。

3) 强调状语

若被强调的部分是状语时，只能用 that 而不能用 when 或 where。请看下例：

【例】It was last year that he took over this company.

他是在去年接管了这家公司的。

【例】It was in 1993 that he began to set up this private company.

正是在 1993 年他开始建立这家私人企业的。

【例】It was in Shanghai that I first met him.

我第一次遇见他是在上海。

【例】It was not until yesterday that he did the experiment.

直到昨天他才做了那个实验。

【例】It was not until several years later that people would accept his idea.

直到数年之后，人们才开始接受他的观点。

注意：It is（was）… that 句型不能强调谓语。若要强调谓语，需用助动词 do。请看下例：

【例】Tom does look well.

汤姆看上去确实很健康。

【例】Do come earlier next time.

下次一定要早点来。

【例】They did work very hard last Sunday.

他们上星期天确实工作很努力。

Exercises

一、根据课文内容，判断以下叙述的正误。

（1）A database is a collection of organized information.

（2）A relational database is a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways.

（3）A distributed database is one that can be dispersed or replicated at certain points in a network.

（4）An object-oriented programming database is one that is congruent with the data defined in object classes and subclasses.

（5）Databases and database managers are prevalent only in large mainframe systems.

（6）The most typical DBMS is a distributed database management system.

（7）A DBMS can be thought of as a file manager that manages data in databases.

（8）The records make up the columns and the fields make up the table rows.

二、根据课文内容填空。

（1）According to types of content databases can be classified into ____________, ____________, ____________ and images.

（2）The relational database was invented by ______________ in ______________.

（3）SQL stands for ________________________. It is a standard language for making interactive queries from a database and updating a database.

（4）SQL statements are used both for _____________________________ from a relational database and _____________________________.

（5）Queries take the form of a command language that lets you ____________, ____________,

______________, find out the location of data and so forth. There is also ______________.

(6) A database management system (DBMS), sometimes just called ________________, is a program that lets one or more computer users ____________________in a database.

(7) A standard user and program interface is ____________________________. A newer kind of DBMS is ________________________________.

(8) A DDBMS (distributed database management system) is a centralized application that manages _________________________as if it were all stored on the same computer.

(9) In a database table, a field is ________________. Fields are organized into __________.

(10) In a database, a record is a group of fields within __________that are relevant to ________ __________.

三、用 it 改写下列句子。

(1) I am very pleased to hear that you have learned computer.

(2) You don't need to worry. It's really unnecessary.

(3) Tom repaired the printer. That was very clear.

(4) To buy this old kind of computer is no longer possible.

(5) He said he didn't like DOS but Windows.

(6) Viruses are the things that worry most computer men.

(7) Is typing a difficult job?

(8) Learning BASIC language or FoxBASE is not particularly hard.

(9) She found that finding out the bug in the program is very tiring.

(10) This work was actually finished yesterday.

四、选择与以下各条叙述意义最接近的词汇。

(1) The type of computer processing where the user of the system communicates directly with the system to input data and instructions and receive output.

(2) The boundary between two systems; a shared boundary between two systems.

(3) The capability of having two or more jobs in the computer at the same time. Execution of the program is interleaved so that in a time interval each job will have been (partly) processed. Processing is not simultaneous.

(4) A pictorial representation of processes and procedures for operation on data. A diagram that describes documents, procedures, processes, and equipment used in processing data in a specific application.

(5) Performing tests and checks on input to ensure that the input operation is legal and that the input it self is correct. Pertaining to a wide variety of tests that can be applied to ensure the correctness of data being input to a computer system.

供选择的答案：

A. decision table

B. environment

C. flowchart

D. input/output system

E. input validation

F. integrated circuit

G. interactive computing

H. interface

I. Multiprogramming

五、听短文，在画线处填写所听到的单词或词组。

A computer database is a structured ___1___ of records or data that is stored in a computer system. The ___3___ is achieved by organizing the data according to a ___2___. The model in most common use today is the ___4___. Other models such as the hierarchical model and the ___5___ use a more explicit representation of relationships (see below for explanation of the various database models).

A computer database relies upon software to ___6___ the storage of data. This software is known as a ___7___ (DBMS). Database management systems are ___8___ according to the database model that they support. The model tends to determine the ___9___ that are available to access the database. A great deal of the internal engineering of a DBMS, however, is independent of the data model, and is concerned with managing factors such as ___10___, concurrency, integrity, and recovery from hardware failures. In these areas there are large differences between products.

六、计算机软件水平考试真题自测（高级程序员级）：选择填空。

___1___ analysis emphasizes the drawing of pictorial system models to document and validate both existing and/or proposed systems. Ultimately, the system models become the ___2___ for designing and constructing an improved system. ___3___ is such a technique. The emphasis in this technique is process-centered. Systems analysts draw a series of process models called ___4___. ___5___ is another such technique that integrates data and process concerns into constructs called objects.

供选择的答案：

1	A. Prototyping	B. Accelerated	C. Model-driven	D. Iterative
2	A. image	B. picture	C. layout	D. blueprint
3	A. Structured analysis		B. Information Engineering	
	C. Discovery Prototyping		D. Object-Oriented analysis	
4	A. PERT	B. DFD	C. ERD	D. UML
5	A. Structured analysis		B. Information Engineering	
	C. Discovery Prototyping		D. Object-Oriented analysis	

Skill Training

计算机英文论文引言写作

引言也称为前言、导言或绪论，它简要介绍本篇论文并提出中心论点。主要由以下各部分组成。

1. 问题提出

引言中问题提出的关键在于要陈述研究动机、目的及贡献。首先，扣紧研究主题，然后提出相关问题是什么？有几个问题？每个问题分别是什么？其次，界定问题，指明问题的背景与内涵（欲解决何种问题）、研究范围（指出研究的边界与限制条件）等。然后，说明本文的研究要达到什么目标，解决什么问题。最后，陈述自己的贡献。评价自己的贡献时，主要在于创新点。此时，不可使用诸如“首次发现”、“填补空白”、“达到世界领先水平”这样的语句。即便是创造性的工作，也应行文谨慎。例如：

To the author's knowledge …

There is little information available in literature about …

Until now we possess information concerning …

Until recently there is some lack of knowledge about …

2. 文献评论

文献评论对前人已经进行的研究给出简要综述并点评，对前人的研究途径、方法进行评判，说明其优缺点。此部分主要包括：概述（归类）、摘要、批判及建议。

3. 研究途径及研究方法

研究途径（approach）是指选择问题和相关数据的标准，以及拟从何种角度切入去探讨该主题与相关问题。

研究方法（method）是指搜集与处理数据的程序与手段，针对自己欲探讨的主题和相关问题拟如何进行搜集和分析资料。研究方法主要分为两类：其一，使用前人提出的方法，即把现有的某种方法用到本文的研究中；其二，对原有方法加以修改与改进，或者提出全新的方法。

应该先决定采用何种研究途径（approach），然后再决定采用何种研究方法（method）。

4. 常用句型

（1）回顾某领域已取得的研究结果或介绍相关知识。

- …is presented in this paper.
- This paper reviews/ outlines …
- This article summarizes …

(2) 阐明论文写作和研究的目的。

- The purpose of this study is to …
- The paper attempts to …
- The study is aiming at …
- The primary goal of this research is …
- Based on recent research the author intends to …
- The authors are now initiating some experimental investigations to …

(3) 陈述论文的论点和作者观点。

- This paper presents/reports/explains the mathematical model and its algorithm used for …
- The calibration and experiment design of multivariate force sensors are discussed/described in this article.
- This paper is mainly devoted to …
- We concern the assessments of the new method for …
- This paper deals with/concerns … for the first time.

(4) 介绍研究过程和研究范围。

- … was investigated/studied/discussed/proposed/stated/identified.
- The paper analyzes the possibility of…
- We study the one-step-synthesis method for… in this paper.
- The study identifies some procedures for…
- This article outlines…
- The scope of study covers…
- Our study includes…
- The paper contains the specific topics/fields on…

Reading Material

Data Mining[1]

1. General Outline of Data Mining

Data Mining is an analytic process designed to explore data (usually large amounts of data, typically business or market related) in search of consistent patterns[2] and/or systematic relationships between variables, and then to validate[3] the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction and predictive data mining is the most common type of data mining and one that has the most direct business applications. The

[1] data mining 数据挖掘
[2] pattern ['pætən] *n.* 模式，式样
[3] validate ['vælideit] *vt.* 证实，验证

process of data mining consists of three stages: ① the initial exploration[1], ② model building or pattern identification with validation/verification, and ③ deployment (i.e., the application of the model to new data in order to generate predictions).

Stage 1: Exploration. This stage usually starts with data preparation which may involve cleaning data[2], data transformations[3], selecting subsets of records and, in case of data sets with large numbers of variables ("fields"), performing some preliminary feature selection operations to bring the number of variables to a manageable range (depending on the statistical methods[4] which are being considered). Then, depending on the nature of the analytic problem, this first stage of the process of data mining may involve anywhere between a simple choice of straightforward[5] predictors for a regression[6] model, to elaborate exploratory analyses using a wide variety of graphical and statistical methods in order to identify the most relevant variables and determine the complexity and/or the general nature of models that can be taken into account[7] in the next stage.

Stage 2: Model building[8] and validation. This stage involves considering various models and choosing the best one based on their predictive performance (i.e., explaining the variability in question and producing stable[9] results across samples). This may sound like a simple operation, but in fact, it sometimes involves a very elaborate process. There are a variety of techniques developed to achieve that goal, many of which are based on so-called "competitive evaluation of models", that is, applying different models to the same data set and then comparing their performance to choose the best. These techniques, which are often considered the core of predictive data mining, include: Bagging (Voting, Averaging), Boosting, Stacking (Stacked Generalizations) and Meta-Learning[10].

Stage 3: Deployment. That final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

The concept of Data Mining is becoming increasingly[11] popular as a business information management tool where it is expected to reveal knowledge structures that can guide decisions in conditions of limited certainty. Recently, there has been increased interest in developing new analytic techniques specifically designed to address the issues relevant to business Data Mining (e.g., Classification[12] Trees), but Data Mining is still based on the conceptual principles of statistics including the traditional Exploratory Data Analysis (EDA) and modeling and it shares with them both some components of its general approaches and specific techniques.

However, an important general difference in the focus and purpose between Data Mining and the traditional Exploratory Data Analysis (EDA) is that Data Mining is more oriented towards

---

[1] exploration [ˌeksplɔːˈreiʃən] *n.* 探测
[2] cleaning data 清洗数据
[3] data transformation 数据变换
[4] statistical method 统计方法
[5] straightforward [streitˈfɔːwəd] *adj.* 简单的，易懂的
[6] regression [riˈgreʃən] *n.* 回归
[7] take into account 重视，考虑
[8] model building 建模
[9] stable [ˈsteibl] *adj.* 稳定的
[10] Meta-Learning 元学习
[11] increasingly [inˈkriːsiŋli] *adv.* 越来越多地，逐渐增加地，日益地
[12] classification [ˌklæsifiˈkeiʃən] *n.* 分类，分级

applications than the basic nature of the underlying phenomena[1]. In other words, Data Mining is relatively less concerned with[2] identifying the specific relations between the involved variables. For example, uncovering the nature of the underlying functions or the specific types of interactive, multivariate[3] dependencies between variables are not the main goal of Data Mining. Instead, the focus is on producing a solution that can generate useful predictions. Therefore, Data Mining accepts among others a "black box" approach to data exploration or knowledge discovery[4] and uses not only the traditional Exploratory Data Analysis (EDA[5]) techniques, but also such techniques as Neural Networks[6] which can generate valid predictions but are not capable of identifying the specific nature of the interrelations between the variables on which the predictions are based.

Data Mining is often considered to be "a blend[7] of statistics, AI (artificial intelligence), and data base research", which until very recently was not commonly recognized as a field of interest for statisticians, and was even considered by some "a dirty word in Statistics". Due to its applied importance, however, the field emerges as a rapidly growing and major area, also in statistics, where important theoretical advances are being made.

2. Models for Data Mining

In the business environment, complex data mining projects may require the coordinate efforts of various experts, stakeholders, or departments throughout an entire organization. In the data mining literature, various "eneral frameworks" have been proposed to serve as blueprints for how to organize the process of gathering[8] data, analyzing data, disseminating results, implementing results, and monitoring improvements.

One such model, CRISP (Cross-Industry Standard Process for data mining) was proposed in the mid-1990s by a European consortium of companies to serve as a non-proprietary standard process model for data mining. This general approach postulates the following, perhaps not particularly controversial[9], general sequence of steps for data mining projects:

Business Understanding ⟷ Data Understanding

↓

Data Preparation ⟷ Modeling

↓

Evaluation

↓

Deployment

---

[1] phenomena [fiˈnɔminə] *n.* 现象

[2] concern with 使关心

[3] multivariate [ˌmʌltiˈveəriit] *adj.* 多变量的，多元的

[4] knowledge discovery 知识发现

[5] EDA (Exploratory Data Analysis) 探测数据分析

[6] Neural Network 神经元网络

[7] blend [blend] *vt.* &n. 混和

[8] gather [ˈgæðə] *n.* 集合，聚集
vi. 聚集，收集，采集

[9] controversial [ˌkɔntrəˈvəːʃəl] *adj.* 争论的，争议的

Another approach, the Six Sigma methodology[1], is a well-structured, data-driven[2] methodology for eliminating defects, waste, or quality control problems of all kinds in manufacturing, service delivery, management, and other business activities. This model has recently become very popular due to its successful implementations in various American industries, and it appears to gain favor worldwide. It postulated a sequence of, so-called, DMAIC steps—Define ⟶ Measure ⟶ Analyze ⟶ Improve ⟶ Control — that grew up[3] from the manufacturing, quality improvement, and process control traditions and is particularly well suited to production environments, including "production of services", i.e., service industries.

Another framework of this kind, actually somewhat similar to Six Sigma, is the approach proposed by SAS Institute called SEMMA—Sample ⟶ Explore ⟶ Modify ⟶ Model ⟶ Assess—which is focusing more on the technical activities typically involved in a data mining project.

All of these models are concerned with the process of how to integrate data mining methodology into an organization, how to "convert data into information", how to involve important stakeholders, and how to disseminate[4] the information in a form that can easily be converted by stakeholders into resources for strategic decision[5] making.

Some software tools for data mining are specifically designed and documented to fit into[6] one of these specific frameworks.

The general underlying philosophy of StatSoft's STATISTICA Data Miner is to provide a flexible data mining workbench[7] that can be integrated into any organization, industry, or organizational culture, regardless of the general data mining process-model that the organization chooses to adopt. For example, STATISTICA Data Miner can include the complete set of (specific) necessary tools for ongoing company wide Six Sigma quality control[8] efforts, and users can take advantage of its (still optional) DMAIC-centric user interface for industrial data mining tools. It can equally well be integrated into ongoing marketing research, CRM (Customer Relationship Management)[9] projects, etc. that follow either the CRISP or SEMMA approach, it fits both of them perfectly well without favoring either one. Also, STATISTICA Data Miner offers all the advantages of a general data mining oriented "development kit[10]" that includes easy to use tools for incorporating into your projects not only such components as custom database gateway solutions, prompted interactive queries, or proprietary algorithms, but also systems of access privileges, workgroup management, and other

[1] Six Sigma methodology 六西格玛方法论
[2] data-driven 数据驱动的
[3] grow up 长大，增长，崛起
[4] disseminate [di'semineit] *vt.* 散布，传播
[5] strategic decision 战略决策
[6] fit into 适合
[7] workbench ['wəːkbentʃ] *n.* 工作台
[8] quality control 质量管理
[9] CRM (Customer Relationship Management) 客户关系管理
[10] development kit 开发工具包

collaborative work tools that allow you to design large scale, enterprise-wide systems that involve your entire organization.

参 考 译 文

数据库基本概念

1. 数据库

数据库是信息的集合，这些信息被组织起来以便可以容易地访问、管理和更新。数据库可以按照其内容分为以下几类：书籍目录数据库、全文本数据库、数字数据库和图像数据库。

在计算领域中，数据库有时也按照其组织方法来分类。当前最流行的方法就是关系数据库，即一个定义数据的以便可以用多种不同的方法来重新组织和访问的表格式数据库。分布式数据库是一个在网络中许多不同的地方分布或复制的数据库。面向对象编程数据库是一个适合用对象类和子类定义数据的数据库。

计算机数据库通常包含数据记录或文件的集合，如销售业务、产品目录和库存以及客户概况。通常，数据库管理程序给用户提供控制读/写访问、产生报表和分析使用情况的能力。数据库和数据库管理程序在大型机系统中非常普遍，但也出现在更小的分布式工作站和中等规模的系统中，如出现在 AS/400 或个人计算机中。

2. 关系数据库

关系数据库是数据项的集合，这些数据项组织为正式描述的表格的一个集合，其中的数据可以用多种方式访问或调整而无须重新组织数据库表。关系数据库由 E. F. Codd 于 1970 年在 IBM 公司首创。

关系数据库的标准用户和应用程序接口是结构化查询语言（SQL）。SQL 语句既可用于对关系数据库进行交互式信息查询也可用于收集报表信息。

除了相对容易建立和访问之外，关系数据库的主要优点是容易扩展。建立了原始数据库后，可以增加新的数据库类别而无须对现有所有应用进行修改。

关系数据库是包含预设种类中数据的表格的集合。每个表（有时也叫作关系）按列包含一个或多个数据类。每行包括由列所定义的类型的唯一数据项。例如，一个典型的商务定单项数据库可以包括一个描述客户的表，该表列有客户姓名、地址、电话号码，等等。另一个表描述订单：产品、客户、日期、销售价格，等等。该数据库的用户可以获得他所需要的数据库概况。一个分部经理也许需要在某个日期之后购买产品的全部客户的概况或报表。同一公司的金融服务经理可以从同一表中获得需要支付的账号报表。

建立一个关系数据库后，可以在一个数据列中定义可能值的域以及未来可以应用到这些值的约束。例如，一个潜在客户域最多可以允许有十个客户的名称，但限制在一个表中只能列出三个这样的客户。

关系数据库的定义会产生一个元数据表或对该表、列、域和约束的正式描述。

3. SQL

SQL 是一种标准语言，用来进行对数据库的交互式查询并更新数据库，如 IBM 的 DB2、微软的 Access，以及来自 Oracle、Sybase 和 Computer Associates 的数据库产品。尽管 SQL 既是一个 ANSI 标准，也是一个 ISO 标准，但许多产品支持对标准语言的专门扩展的 SQL。请求的形式是命令行语言，可以让你进行选择、插入、更新、找出数据的位置，等等，同时也有一个编程接口。

4. 数据库管理系统

数据库管理系统（DBMS）有时也叫作数据库管理器，是让一个或多个计算机用户建立和访问数据库中数据的程序。DBMS 管理用户查询（及来自其他程序的查询），这样用户和其他程序就不需要知道这些数据在介质中存储的物理位置，并且在多用户系统中，也不必知道还有谁可能正在访问这些数据。在处理用户查询时，DBMS 确保数据的完整性（也就是，确保可以持续地被访问并且一直按照预先要求组织好）和安全性（确保只有那些有访问权的用户才可以访问这些数据）。最典型的 DBMS 是关系数据库管理系统（RDBMS）。一个标准的用户和程序接口是 SQL。一个更新的 DBMS 是面向对象数据库管理系统（ODBMS）。

DBMS 可以被看作一个文件管理器，它管理数据库中的数据而不是文件系统中的文件。在 IBM 的大型机操作系统中，非关系数据管理器曾经（并且现在也是，因为这些老的应用系统仍然在使用）以访问方法而知名。

DBMS 通常是数据库产品的固有部分。在 PC 上，微软的 Access 是单一或小组用户 DBMS 的一个流行范例。微软的 SQL Server 是适用于多用户（客户）数据库查询的一个范例。其他流行的 DBMS（顺便说一下，这些全部都是 RDBMS）是 IBM 的 DB2、Oracle 的数据库管理产品线以及 Sybase 的产品。

IBM 的信息管理系统（IMS）是最初的 DBMS 之一。DBMS 也可被像 IBM 的客户信息管理系统（CICS）这样的业务管理程序使用，或与其结合使用。

5. 分布式数据库

分布式数据库是数据库中的某些部分存储在网络中的多个计算机中的数据库。用户可以在他们自己的位置访问该数据库的一部分，这样他们可以访问与其工作相关的数据而不会影响其他人的工作。

6. DDBMS

DDBMS（分布式数据库管理系统）是一个集中应用程序，管理一个分布式数据库，就像该数据库存储在同一计算机上一样。DDBMS 定期地保证所有数据的同步，并且在多个用户必须访问相同数据的情况下，确保在一个地方对数据的更新和删除在其他地方存储的数据中会自动反映出来。

7. 字段

在数据库表中，字段是用于单一数据块的数据结构。字段组成为记录，包括表中与特定

实体相关的全部信息。例如，在一个叫作客户联系信息的表中，电话号码可能是一行中的一个字段，该行也包含了其他字段，如街道地址和城市。记录构成了表行而字段构成了表列。

8. 记录

在数据库中，记录（有时也叫作行）是表中与一个特定实体相关的一组字段。例如，在一个叫作客户联系信息的表中，一行可能包括这样的字段：标识号、名字、街道地址、城市、电话号码，等等。

9. 表

在关系数据库中，表（有时也叫作文件）把单一主题的信息组成为行和列。例如，一个商用数据库通常包括一个客户信息表，该表把客户账号、地址、电话号码等存储为一系列的列。每个单一的数据块（如账号）是表中的一个字段。一列由单一字段的全部实体组成，如全部客户的电话号码。字段依次地被组织为记录，这就组成了信息的完整集合（如某一特定客户的信息集合），每个记录构成一行。这个规范处理过程决定了怎样将数据最有效地组织为表。

Lesson 9

Text

Cloud Computing

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). The name cloud computing was inspired by the cloud symbol that's often used to represent the Internet in flowcharts and diagrams.

A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic—a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing.

1. Cloud

A cloud can be private or public. A public cloud sells services to anyone on the Internet. (Currently, Amazon Web Services is the largest public cloud provider.) A private cloud is a proprietary network or a data center that supplies hosted services to a limited number of people. When a service provider uses public cloud resources to create their private cloud, the result is called a virtual private cloud. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services.

1.1 Public Cloud

A public cloud is one based on the standard cloud computing model, in which a service provider makes resources, such as applications and storage, available to the general public over the Internet. Public cloud services may be free or offered on a pay-per-usage model.

The main benefits of using a public cloud service are:

- Easy and inexpensive set-up because hardware, application and bandwidth costs are covered by the provider.
- Scalability to meet needs.
- No wasted resources because you pay for what you use.

Examples of public clouds include Amazon Elastic Compute Cloud (EC2), IBM's Blue Cloud, Sun Cloud, Google AppEngine and Windows Azure Services Platform.

1.2 Private Cloud

Private cloud (also called internal cloud or corporate cloud) is a marketing term for a proprietary computing architecture that provides hosted services to a limited number of people behind a firewall.

Advances in virtualization and distributed computing have allowed corporate network and datacenter administrators to effectively become service providers that meet the needs of their "customers" within the corporation.

Marketing media that uses the words "private cloud" is designed to appeal to an organization that needs or wants more control over their data than they can get by using a third party hosted service such as Amazon's Elastic Compute Cloud or Simple Storage Service.

1.3 Community Cloud

Community cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third party and hosted internally or externally. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized.

1.4 Hybrid Cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models. Such composition expands deployment options for cloud services, allowing IT organization to use public cloud computing resources to meet temporary needs. This capability enables hybrid clouds to employ cloud bursting for scaling across clouds.

Cloud bursting is an application deployment model in which an application runs in a private cloud or data center and "bursts" to a public cloud when the demand for computing capacity increases. A primary advantage of cloud bursting and a hybrid cloud model is that an organization only pays for extra compute resources when they are needed.

Cloud bursting enables data centers to create an in-house IT infrastructure that supports average workloads, and use cloud resources from public or private clouds, during spikes in processing demands.

By utilizing "hybrid cloud" architecture, companies and individuals are able to obtain degrees of fault tolerance combined with locally immediate usability without dependency on internet connectivity. Hybrid cloud architecture requires both on-premises resources and off-site (remote) server-based cloud infrastructure.

Hybrid clouds lack the flexibility, security and certainty of in-house applications. Hybrid cloud

provides the flexibility of in house applications with the fault tolerance and scalability of cloud based services.

2. What is XaaS (anything as a service)?

XaaS is a collective term said to stand for a number of things including "X as a service", "anything as a service" or "everything as a service". The acronym refers to an increasing number of services that are delivered over the Internet rather than provided locally or on-site. XaaS is the essence of cloud computing.

The most common examples of XaaS are SaaS, IaaS and PaaS. The combined use of these three is sometimes referred to as the SPI (SaaS, PaaS, IaaS) model. Other examples of XaaS include Storage as a Service (SaaS), Communications as a Service (CaaS), Network as a Service (NaaS) and Monitoring as a Service (MaaS).

2.1 IaaS

IaaS is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. The client typically pays on a per-use basis.

Characteristics and components of IaaS include:

- Utility computing service and billing model.
- Automation of administrative tasks.
- Dynamic scaling.
- Desktop virtualization.
- Policy-based services.
- Internet connectivity.

IaaS is one of three main categories of cloud computing service. The other two are SaaS and PaaS.

Infrastructure as a Service is sometimes referred to as Hardware as a Service (HaaS).

2.2 PaaS

PaaS is a way to rent hardware, operating systems, storage and network capacity over the Internet. The service delivery model allows the customer to rent virtualized servers and associated services for running existing applications or developing and testing new ones.

PaaS is an outgrowth of SaaS, a software distribution model in which hosted software applications are made available to customers over the Internet. PaaS has several advantages for developers. With PaaS, operating system features can be changed and upgraded frequently. Geographically distributed development teams can work together on software development projects. Services can be obtained from diverse sources that cross international boundaries. Initial and ongoing costs can be reduced by the use of infrastructure services from a single vendor rather than

maintaining multiple hardware facilities that often perform duplicate functions or suffer from incompatibility problems. Overall expenses can also be minimized by unification of programming development efforts.

On the downside, PaaS involves some risk of "lock-in" if offerings require proprietary service interfaces or development languages. Another potential pitfall is that the flexibility of offerings may not meet the needs of some users whose requirements rapidly evolve.

2.3 SaaS

SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet.

SaaS is becoming an increasingly prevalent delivery model as underlying technologies that support Web services and mature service-oriented architecture (SOA) as well as new developmental approaches, such as Ajax, become popular. Meanwhile, broadband service has become increasingly available to support user access from more areas around the world.

SaaS is closely related to the ASP (Application Service Provider) and on demand computing software delivery models. IDC (International Data Corporation) identifies two slightly different delivery models for SaaS. The hosted application management (hosted AM) model is similar to ASP: a provider hosts commercially available software for customers and delivers it over the Web. In the software on demand model, the provider gives customers network-based access to a single copy of an application created specifically for SaaS distribution.

Benefits of the SaaS model include:

- easier administration.
- automatic updates and patch management.
- compatibility (all users will have the same version of software).
- easier collaboration, for the same reason.
- global accessibility.

The traditional model of software distribution, in which software is purchased for and installed on personal computers, is sometimes referred to as software as a product.

New Words

deliver	[diˈlivə]	*vt.* 发布，交付
host	[həust]	*vt.* 托管
infrastructure	[ˈinfrəˌstrʌktʃə]	*n.* 基础设施
inspired	[inˈspaiəd]	*vt.* 作为……的灵感来源，启迪，启发
flowchart	[ˈfləutʃɑːt]	*n.* 流程图，程序框图
distinct	[disˈtiŋkt]	*adj.* 清楚的，明显的，截然不同的，独特的
elastic	[iˈlæstik]	*adj.* 弹性的

significant	[sigˈnifikənt]	*adj.* 有意义的，重大的，重要的
virtualization	[ˌvəːtjuəliˈzeiʃən]	*n.* 虚拟化
accelerate	[əkˈseləreit]	*vt.* 加速，加快
proprietary	[prəˈpraiətəri]	*adj.* 私有的
scalability	[ˌskeiləˈbiliti]	*n.* 可扩展性，可量测性
scalable	[ˈskeiləbl]	*adj.* 可升级的
hybrid	[ˈhaibrid]	*adj.* 混合的
		n. 混合物
maintain	[menˈtein]	*vt.* 维持，维修，维护
datacenter	[ˈdeitəˈsentə]	*n.* 数据中心，资料处理中心
jurisdiction	[ˌdʒuərisˈdikʃən]	*n.* 权限
potential	[pəˈtenʃəl]	*adj.* 潜在的，可能的
composition	[kɔmpəˈziʃən]	*n.* 合成物
temporary	[ˈtempərəri]	*adj.* 暂时的，临时的
employ	[imˈplɔi]	*v.* 使用
burst	[bəːst]	*v.* 爆裂，爆发
		n. 爆发，脉冲
extra	[ˈekstrə]	*adj.* 额外的
		adv. 特别地，非常，另外
workload	[ˈwəːkləud]	*n.* 工作量
utilize	[juːˈtilaiz]	*vt.* 利用
usability	[ˌjuːzəˈbiləti]	*n.* 可用性
dependency	[diˈpendənsi]	*n.* 依靠，信赖
outsource	[ˈəutˌsɔːs]	*n.* 外包
connectivity	[kənekˈtiviti]	*n.* 连通性
collective	[kəˈlektiv]	*adj.* 集体的，共同的，集合的
essence	[ˈesens]	*n.* 基本，本质
provision	[prəˈviʒən]	*n.* 供应；预备，防备，规定
outgrowth	[ˈautgrəuθ]	*n.* 派出，结果，副产物
upgrade	[ˈʌpgreid]	*n.* 升级，上升
		vt. 使升级，提升
project	[ˈprɔdʒekt]	*n.* 项目，工程
diverse	[daiˈvəːs]	*adj.* 不同的，变化多的
facility	[fəˈsiliti]	*n.* 设备，工具
duplicate	[ˈdjuːplikeit]	*adj.* 复制的，副的；两重的，两倍的，完全相同的
		n. 复制品，副本
		vt. 复制；使加倍，使成双
incompatibility	[ˈinkəmˌpætəˈbiliti]	*n.* 不兼容性

expense	[ikˈspens]	*n*. 费用，代价，损失，开支
unification	[ˌjuːnifiˈkeiʃən]	*n*. 统一，合一，一致
pitfall	[ˈpitfɔːl]	*n*. 缺陷
evolve	[iˈvɔlv]	*v*.（使）发展，（使）进展
mature	[məˈtjuə]	*adj*. 成熟的，到期的
		vt. 使成熟
		vi. 成熟，到期
meanwhile	[ˈmiːnwail]	*n*. 其间，其时（= meantime）
slightly	[ˈslaitli]	*adv*. 稍微地
automatic	[ˌɔːtəˈmætik]	*adj*. 自动的，机械的
collaboration	[kəˌlæbəˈreiʃən]	*n*. 协作，合作
accessibility	[ˌækəsesiˈbiliti]	*n*. 可访问性

Phrases

cloud computing	云计算
divide into	分成
Infrastructure-as-a-Service（IaaS）	基础设施即服务
Platform-as-a-Service（PaaS）	平台即服务
Software-as-a-Service（SaaS）	软件即服务
cloud symbol	云符号
distributed computing	分布计算
public cloud	公共云
hybrid cloud	混合云
private cloud	私有云，专有云
virtual private cloud	虚拟私有云
third party	第三方
community cloud	社区云，社团云
spread over	遍布在……，分布在……
be able to	能，会
fault tolerance	容错
combine with…	与……结合
on-premises resource	本地资源，本处资源
Storage as a Service（SaaS）	存储即服务
Communications as a Service（CaaS）	通信即服务
Network as a Service（NaaS）	网络即服务
Monitoring as a Service（MaaS）	监控即服务
be responsible for	为……负责，形成……的原因

Hardware as a Service（HaaS）	硬件即服务
development language	开发语言，编程语言
distribution model	发布模式
suffer from	忍受，遭受
Service-Oriented Architecture（SOA）	服务导向架构
Application Service Provider（ASP）	应用服务提供商
on demand	按需
patch management	修补管理

Abbreviations

XaaS（anything as a service）	万事即服务，一切皆服务
SPI（SaaS，PaaS，IaaS）	软件即服务，平台即服务，基础设施即服务
IDC（International Data Corporation）	国际数据公司
AM（Application Manage）	应用管理

Notes

［1］Private or public，the goal of cloud computing is to provide easy，scalable access to computing resources and IT services.

本句中，private or public 作让步状语，可以扩展为 whether the cloud is private or public. 意思是“无论私有云还是公共云”。请看下例：

Big or small，the company should pay tax.

无论公司大小，都应该纳税。

［2］A public cloud is one based on the standard cloud computing model，in which a service provider makes resources，such as applications and storage，available to the general public over the Internet.

本句中，in which a service provider makes resources，such as applications and storage，available to the general public over the Internet 是一个非限定性定语从句，对 the standard cloud computing model 作进一步补充说明。在该从句中 such as applications and storage 是插入语，对 resources 进行举例说明。

［3］Private cloud（also called internal cloud or corporate cloud）is a marketing term for a proprietary computing architecture that provides hosted services to a limited number of people behind a firewall.

本句中，（also called internal cloud or corporate cloud）对 private cloud 进行解释说明。that provides hosted services to a limited number of people behind a firewall 是一个定语从句，修饰和限定 a proprietary computing architecture。

［4］Marketing media that uses the words "private cloud" is designed to appeal to an organization that needs or wants more control over their data than they can get by using a third party

hosted service such as Amazon's Elastic Compute Cloud or Simple Storage Service.

本句中，that uses the words "private cloud" 是一个定语从句，修饰和限定 Marketing media。that needs or wants more control over their data than they can get by using a third party hosted service such as Amazon's Elastic Compute Cloud or Simple Storage Service 也是一个定语从句，修饰和限定 an organization。在该定语从句中，by using a third party hosted service such as Amazon's Elastic Compute Cloud or Simple Storage Service 作方式状语。appeal to 的意思是“取悦；吸引”。

[5] Cloud bursting is an application deployment model in which an application runs in a private cloud or data center and "bursts" to a public cloud when the demand for computing capacity increases.

本句中，in which an application runs in a private cloud or data center and "bursts" to a public cloud when the demand for computing capacity increases 是一个定语从句，修饰和限定 an application deployment model。when the demand for computing capacity increases 是一个状语从句，修饰"bursts"。

Grammar

被 动 语 态

语态是动词的一种形式，它表示主语和谓语的不同关系。语态有两种：主动语态和被动语态。

主动语态表示主语是动作的执行者，请看下例：

【例】He designed this building.

他设计了这座大楼。

被动语态表示主语是动作的承受者，请看下例：

【例】This building was designed by him.

这座大楼是他设计的。

1. 被动语态的构成

在科技英语中，被动语态的应用特别多。被动语态的构成如下：

主语 + be + 及物动词的过去分词

值得注意的是，只有及物动词才能构成被动语态，不及物动词和系动词没有被动语态。

2. 科技英语中主要时态的被动语态形式

1）一般现在时

被动语态构成如下：

主语 + am（is，are）+ 及物动词的过去分词

【例】The manager's office is cleaned every day.

经理的办公室每天都有人打扫。

【例】You are wanted on the phone.

有人给你打电话。

【例】I am asked to solve this problem by him.

他请我解决这个问题。

2）一般过去时

一般过去时被动语态构成如下：

主语 + was（were）+及物动词的过去分词

【例】These computers were made last year.

这些计算机是去年制造的。

【例】When was this printer bought?

这台打印机是什么时候买的?

【例】That plotter was not bought in Shanghai.

那台绘图机不是在上海买的。

3）一般将来时

一般将来时被动语态构成如下：

主语 + will be + 及物动词的过去分词

当主语是第一人称时，可用如下构成：

主语 + shall be + 及物动词的过去分词

【例】When will this building be completed?

这座大楼何时竣工?

【例】What tools will be needed to solve this tough problem? PCTOOLS.

要解决这个难题需要什么工具? PCTOOLS。

【例】He will not be allowed to do it.

不会让他做这件事的。

4）现在进行时

现在进行时被动语态构成如下：

主语 + am（is，are）being + 及物动词的过去分词

【例】Laser printers are being produced in that factory.

那个工厂正在制造激光打印机。

【例】Our pin printer is being repaired by Tom.

汤姆正在修理我们的针式打印机。

5）过去进行式

过去进行式被动语态构成如下：

主语 + was（were）being + 及物动词的过去分词

【例】The laboratory building was being built then.

实验大楼当时正在建造。

【例】Were the computers being tested?

当时正在验机吗?

【例】It was not the problem of men but of money that was being discussed.

当时正在讨论的不是人力问题，而是财力。

6）现在完成时

现在完成时被动语态构成如下：

主语 + have（has）been + 及物动词的过去分词

【例】This book has been translated into several languages.

这本书已被翻译成好几种语言。

【例】Has the designed been finished?

设计做完了吗?

【例】The letter hasn't been posted.

信还没有寄出。

7）过去完成时

过去完成时被动语态构成如下：

主语 + had been + 及物动词的过去分词

【例】When he came back, the problem had already been solved.

他回来时，问题已经解决了。

【例】Production costs had been greatly reduced.

生产成本大大地降低了。

【例】Had the desktop scanner been repaired?

台式扫描仪当时被修理过了吗?

注意：完成进行时和将来进行时没有被动语态。若这类时态的主动结构要变为被动结构，可用完成时态或一般时态。但很少用到被动语态的将来完成时和过去将来完成时。

3. 常用被动语态的几种情况

（1）当强调的是动作的承受者或给动作的承受者较大关注时，多用被动语态。这时，由于动作的执行者处于次要地位，句子中 by 引导的短语可以省略。请看下例：

【例】This desk is made of wood.

这张课桌是用木头制成的。

【例】The virus in the computer has been found out.

计算机中的病毒已经找出来了。

（2）当不知道或不想说出动作的执行者时，可使用被动语态。这时句子中不带由 by 引导的短语。请看下例：

【例】The glass was broken last Friday.

玻璃是上周五打破的。

【例】Such books are written for undergraduates.

这些书是为大学生写的。

【例】I was told to go there at once.

我被告知马上到那去。

（3）当动作的执行者是“物”而不是“人”时，常用被动语态。请看下例：

【例】This machine is controlled by a computer.

这台机器由计算机控制。

【例】All the work in this factory is done by two robots.

这家工厂里所有的工作都是由两个机器人完成的。

（4）当动作的执行者已为大家所熟知而没有必要说出来时，也常常使用被动语态。请看下例：

【例】This kind of computer was used several years ago.

这种计算机是几年前使用的。

【例】The laboratory building was completed last year.

这座实验大楼是去年竣工的。

【例】This factory was built twenty years ago.

这座工厂是二十年前兴建的。

（5）使用被动语态能更好地安排句子。请看下例：

【例】This plan was supported by those who came from Beijing and wanted to invest money in this company.

此计划得到了那些来自北京并想给这家公司投资的人的支持。

句中，若用主动语态，则会使句子显得头重脚轻，不平衡。

【例】The professor came into the hall and was warmly applauded by the audience.

教授走进大厅，大家热烈鼓掌。

使用被动语态时，只需要一个主语即可。

4. 情态动词的被动语态

情态动词的被动语态构成如下：

主语 + 情态动词 + be + 及物动词的过去分词

请看下例：

【例】This book can't be taken out of the reading-room.

这本书不能带出阅览室。

【例】This work must be finished by two o'clock.

这项工作必须在两点钟之前完成。

【例】These tools may be needed in your work.

工作中可能需要这些工具。

5. 双宾语的被动语态

英语中有些动词可以有两个宾语，即直接宾语和间接宾语。若用直接宾语作被动语态的

主语，则需在间接宾语前加一个介词 to 或 for。若用间接宾语作主语，则仍然把直接宾语放在谓语之后。请看下例：

【例】He showed the manager the plan.

他把计划给经理看。

句中，the manager 是间接宾语，the plan 是直接宾语。本句变为被动语态：

The manager was shown the plan by him.

他把计划给经理看。

The plan was shown to the manager by him.

计划是他给经理看的。

【例】The manager bought her a house.

经理给她买了一座房子。

句中，her 是间接宾语，a house 是直接宾语。本句变为被动语态：

She was bought a house by the manager.

经理给她买了一座房子。

A house was bought for her by the manager.

这座房子是经理给她买的。

注意：把带有两个宾语的句子变为被动语态时，多数都把间接宾语变为主语。这样句子显得更自然一些。

6. 短语动词的被动语态

除了单一的及物动词可用于被动语态外，一些相当于及物动词的动词短语也可以用于被动语态。请看下例：

【例】The design was approved of.

这个设计被通过了。

句中，动词短语是 approve of，意思为“通过”。

【例】Scientific researches are often carried on in this institute.

科研工作经常是在这个研究所进行的。

句中，动词短语是 carry on，意思为“进行”。

【例】All the dirty marks on the products must be done away with.

必须除掉产品上的所有污点。

句中，动词短语是 do away with，意思为“除掉、去掉”。

【例】The properties of this machine should be made full use of.

应该充分利用该机器的性能。

句中，动词短语是 make full use of，意思为“充分利用”。

7. 被动语态的译法

（1）英语中大部分的被动语句在翻译成汉语时，都由“被”、“由”、“为”、“受”、“用”等词来表达。请看下例：

【例】He was praised for what he had done.

他为他所做的事情而受到表扬。

【例】The experiment was done by the students.

这个实验是由这些学生们做的。

【例】He was made monitor of the class.

他被选为班长。

（2）根据汉语习惯，可翻译成主动语句。若没有动作的执行者，可加上“我们”、“人们”、“有人” 等。请看下例：

【例】Many new factories have been built in the south.

在南方人们已经兴建了许多工厂。

【例】He is ordered to take this desk away.

有人命令他把这张课桌搬走。

【例】Computers are found very important nowadays.

如今人们发现计算机很重要。

（3）可把被动句翻译成无人称的主动语句。请看下例：

【例】F1 may be used as a function key.

可把 F1 作为功能键。

【例】Windows are used to solve this problem.

可用 Windows 来解决这个问题。

注意：英语中有些单词用主动表示被动。请看下例：

【例】Your pen writes well.

你的钢笔很好用。

【例】This book sells well.

这本书很畅销。

Exercises

一、根据课文内容，判断以下叙述的正误。

（1）The consumer only needs a personal computer and Internet access to have a cloud service.

（2）More and more people are interested in cloud computing.

（3）A private cloud is a proprietary network or a data center that supplies hosted services to a large number of people.

（4）A public cloud is one based on the standard cloud computing model.

（5）Community cloud can only be managed internally.

（6）Cloud bursting is an application provision model in which an application runs in a private cloud or data center and "bursts" to a public cloud when the demand for computing capacity increases.

（7）SPI model refers to the combined use of SaaS，PaaS，IaaS.

（8）Infrastructure as a Service is a distribution model in which an organization outsources the equipment used to support operations.

(9) Infrastructure as a Service is sometimes referred to as Software as a Service.

(10) Platform as a Service (PaaS) allows development teams in different places work together on software development projects.

二、根据课文内容填空。

(1) The hosted services are broadly divided into three categories. They are ________________, ________________ and ________________.

(2) A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold ______________, typically by the minute or the hour; it is ___________— a user can have as much or as little of a service as they want at any given time; and the service is fully managed by ______________.

(3) The goal of cloud computing is to ____.

(4) Examples of public clouds include __________________, __________________________, ___________________, ___________________ and ___________________.

(5) Private cloud is a marketing term for a proprietary computing architecture that provides ____.

(6) A primary advantage of cloud bursting and a hybrid cloud model is that an organization only ____.

(7) Hybrid cloud architecture requires ___________________ and ___________________.

(8) It is said that XaaS is a collective term which stands for a number of things including _______________, ________________ or ________________. It is ________________ of cloud computing.

(9) The most common examples of XaaS are ___________________, ___________________ and ___________________.

(10) ___________________ is a way to rent hardware, operating systems, storage and network capacity over the Internet.

三、翻译下列句子。

(1) The printer retains to this centering position when the power is turned off.

(2) Your computer is made up of many parts called hardware.

(3) On many systems the floppy disk drive is called A drive and the hard disk drive is called C drive.

(4) If you are sure a unique name has been assigned to the new file, press Enter.

(5) Before being able to work on the network, a person must be designated as a network user.

(6) All network information is stored on the file server's hard disk. The system for storing that information is called the "directory structure".

(7) If errors were discovered, the programmer could halt the program, examine the contents of memory and registers, and debug the program directly from the console.

(8) The formatting process can be sped up by using the [/q] switch.

(9) Windows commands are listed on menus.

(10) FOR…NEXT loops may be nested; that is, one FOR…NEXT loop may be placed inside another FOR…NEXT loop.

四、从供选择的词汇中选择最合适的填在文中相应数字处。

It is traditional when dealing with languages of all sorts to try to separate concerns with ___1___, the subject of syntax, from concerns with ___2___, the field of semantics. Consider the simple "language" of binary numerals. Some examples of binary numerals are

0	1
101	0101
11001	101111

A communication in this language evidently consists of a finite sequence of characters "0"、"1". This is just syntax however, and says ___3___ about what such a communication is intended to mean.

Numbers are "___4___" mathematical concepts, whereas the digit strings that appear on paper are numerals, that is to say, ___5___ representations or descriptions of numbers.

供1、2和3选择的答案:

A. symbol

B. form

C. meaning

D. context

E. nothing

F. concept

供4和5选择的答案:

A. abstract

B. concrete

C. simple

D. ordinary

E. symbolic

F. logic

五、听短文,在画线处填写所听到的单词或词组。

Artificial intelligence (AI) is the intelligence of ___1___ and the branch of computer science which aims to create it.

Major AI textbooks define artificial intelligence as "the study and design of ___2___," where an intelligent agent is a system that perceives its environment and takes actions which maximize its chances of success. John McCarthy, who coined the term in 1956, defines it as "the science and ___3___ of making intelligent machines."

Among the traits that researchers hope machines will exhibit are ___4___, knowledge,

planning, learning, ____5____, perception and the ability to move and manipulate objects. ____6____ (or "strong AI") has not yet been achieved and is a long-term goal of some AI research.

AI research uses tools and insights from many fields, including computer science, psychology, philosophy, neuroscience, cognitive science, linguistics, ontology, operations research, economics, ____7____, probability, optimization and ____8____. AI research also overlaps with tasks such as robotics, control systems, scheduling, ____9____, logistics, ____10____, facial recognition and many others.

六、计算机软件水平考试真题自测（高级程序员级）：选择填空。

Virtualization is an approach to IT that pools and shares ____1____ so that utilization is optimized and supplies automatically meet demand. Traditional IT environments are often silos, where both technology and human ____2____are aligned around an application or business function. With a virtualized ____3____, people, processes, and technology are focused on meeting service levels, ____4____is allocated dynamically, resources are optimized, and the entire infrastructure is simplified and flexible. We offer a broad spectrum of virtualization ____5____that allows customers to choose the most appropriate path and optimization focus for their IT infrastructure resources.

供选择的答案：

1. A. advantages	B. resources	C. benefits	D. precedents
2. A. profits	B. costs	C. resources	D. powers
3. A. system	B. infrastructure	C. hardware	D. link
4. A. content	B. position	C. power	D. capacity
5. A. solutions	B. networks	C. interfaces	D. connections

Skill Training

计算机英文论文段落的写作

文章由段落组成，段落由一群彼此相关的、为了阐述同一中心内容或为了表达同一想法的句子构成。在一个段落中，句与句之间表达的内容必须连贯、流畅，整个段落必须条理分明、合乎逻辑，并按照问题所定范围展开。

1. 构成段落的三大组成部分

典型的英语段落包括一个主题句（topic sentence）和若干扩展句（supporting sentences），必要时还有一个结论句（concluding sentence）。主题句表达的是段落的中心思想，是段落的核心，是为整篇文章的中心服务的；而扩展句则具体地描述、说明、阐述或论证主题句所体现的中心思想。结论句往往是用来强调、重申主题或归纳、总结段落大意。

1.1 主题句

主题句提出本段的命题，它限定了段落的基调。主题突出的主题句也能使读者一目了

然，有利于读者预测下文内容。主题句的作用是要告诉读者该段的主题思想，该段将围绕这个主题思想逐步展开、定义、论述、分类、解释和举例说明等，所以段落主题句中必须包含一个主导思想，这是一个等待发展的思想，也是段落的主旨所在。同时，段落主题句不能写得太笼统，也不能涉及面太窄。一个段落的容量是有限的，如果主题句限定范围太宽，其内容就无法在一个段落中阐述清楚；如果限定范围太窄又不利于段落的发展。段落主题句所限定的内容必须符合段落写作的目的，有助于段落的铺开和抒发。倘若段落的主题句过于笼统或模糊，就无法成为整个段落发展的指南针，就难以合理组织相关材料来进一步描述、说服或阐述主题，甚至使整个段落乃至整篇文章结构松散、逻辑不清。

主题句可居段首，开门见山，一目了然，便于读者掌握段落中心；也可居段尾，制造悬念，便于给读者留下深刻印象；还可居段中，便于作者进行比较或对照。对科技论文写作而言，大部分主题句都放在段首。

1.2 扩展句

扩展句是对主题句所陈述的思想、观点加以展开，提出各种细节或例证以阐述或证明主题。主题句是段落的主干，扩展句是段落的血肉。段落展开方式有以下几种。

1.2.1 时间顺序法（development by time）

按时间的顺序展开段落，读者可以按照时间的脉络清楚地掌握段落中所陈述的内容。

1.2.2 空间顺序法（development by space）

按空间顺序展开段落，通常采用由远及近、由近及远、由上而下、由下而上等方法展开。

1.2.3 事物进程法（development by process）

事物进程法是指按事物的发展过程展开段落，此方式常用于说明文，用来陈述做某事的过程，如生产过程、操作过程或制造过程等。一般按其进行的顺序或时间发生的先后组织材料，简练而准确地介绍各步骤的基本内容。

1.2.4 细节描述法（development by details）

细节描述法，即利用细节描写来展开段落，有时与“举例说明法”(development by example）相似。若细节的运用是为了更详尽地描述，称之为“细节描述法”；若细节的描述是为了阐明一个观点，称之为“举例说明法”。在“举例说明法”中通常运用“for example”,“for instance”等来引出例子。

1.2.5 因果法（development by cause and effect）

人们在思维活动中，常常会根据某事物的原因推导出其结果或根据某种结果分析其原因。写作中“因果法”就是这种思维方式的体现。“因果法”可分为“先因后果”和“先果后因”两种。

此外，段落的展开方式还有“比较对照法”、“分类法”、“定义法”、“概括法”等，在具体的写作中，有时可以只用一种方法，有时也可以综合使用。这样会使自己的观点表述给人留下更深刻的印象，更令人信服。

1.3 结论句

为了求得段落结构的整体性，使段落内容统一，结构紧凑，条理清晰，首尾呼应，一些作者喜欢利用段落的最后一句简要地归纳段落大意，或重申段落主题写结论句。写结论句时，要特别注意以下三点：

（1）不要突然冒出与该段主题不相关甚至相互矛盾的新话题；

（2）不要过于冗长；

（3）语气不能太武断，最好不要使用 certainly，sure，absolutely，must 等，应尽量用委婉、温和的表达法，如 may，would，perhaps，suggest，probably，be likely 等。

2. 段落的两大要素

段落应具有一致性和连贯性两大要素。

2.1 一致性（consistency）

一致性是指段落只能有一个中心思想（controlling idea），段落中所有内容都必须围绕这一中心思想展开。段落中的所有句子都紧扣一个中心，没有任何偏离主题的题外话。主题句的确定与辅助细节的筛选已基本能保证段落的一致性。在一个段落中只能讨论一个主要内容，这个主要内容是由主题句概括叙述，然后通过各个支持句逐一阐述。主题句通常置于段落之首，这符合英语的语篇思维特征，先采用主题句开门见山地摆出问题，随之辅以详细说明。

2.2 连贯性（coherence）

连贯性是指扩展句应该按某种逻辑顺序排列，以便使段落便于阅读和理解；同时，各部分内容应该用适当的过渡词或短语连接起来。连贯性要求段落中句子的排列自然合理，而且句子之间的衔接要合理自然。因此，确定了支持主题句的细节之后，应该认真考虑辅助语句的顺序。

3. 文章的衔接技巧

要做到连贯性，就必须注意句子与句子之间的衔接（cohesion）。文章通过一定的衔接手段，将句与句、段与段有机地组合起来，构成一个完整的语义单位。实现段落连贯的过渡手段主要是词汇或语法手段和逻辑顺序。

3.1 重复（reiteration）

（1）原词的重复，又称为原词复现，指同一主题词或关键词的重复出现。

（2）同义词或近义词的衔接，又称为同义复现，指同义词、近义词重复出现在语篇中，语篇中的句子通过这种关系达到了相互衔接。

3.2 共现（co-occurrence）

共现是指一系列相关的词项在文章中经常共同出现，又称为搭配性衔接（collocation cohesion）。

3.3 替代（substitution）

语篇中的代词构成了替代衔接关系。一般而言，one 指代前面的单数可数名词；that 指代前面的不可数名词或句子；this 指代前面的单数名词或句子；it 指代前面的单数名词或整个句子；they 或 them 指代前面的复数名词。

3.4 用并列结构衔接

并列结构使句子强而有力，清楚明白，增强某段或整篇文章的韵律，使之优雅流畅。

3.5 用词汇或短语衔接

通过适当的词汇或短语来连接。

- 表示平行对等或选择：and，both…and，as well as，neither…nor，or，either…or 等。
- 表示转折：but，yet，while，however，on the contrary，on the other hand 等。
- 表示结果关系：for，so，therefore，as a result，because of，due to，owing to，thanks to 等。
- 表示时间顺序：first，then，later，meanwhile，in the end，finally，after that，since then，for the first time，at last，as soon as，the next moment 等。
- 表示解释说明：that is（to say），in other words，such as，for example，for instance，and so on 等。
- 表示递进关系：what's more/worse，what's better，besides，in addition，worse still，moreover，above all，not only…but also 等。
- 表示总结：in short，in brief，in a word，on the whole，to sum up 等。

综上所述，一个段落应该包含三个组成部分并具有两大要素，即包含主题句、扩展句、结论句和具有一致性、连贯性。同时，要恰当地衔接各个段落，使文章内容连贯，结构完整。

Reading Material

Big Data—What Is It?

1. Definition

Big data[1] is a popular term used to describe the exponential[2] growth, availability and use of information, both structured and unstructured[3]. Much has been written on the big data trend and how it can serve as the basis for innovation, differentiation and growth.

According to IDG[4], it is imperative that organizations and IT leaders focus on the ever-

---

[1] big data 大数据
[2] exponential [ˌekspəuˈnenʃəl] *n.* 指数
adj. 指数的，幂数的
[3] unstructured [ʌnˈstrʌktʃəd] *adj.* 非结构化的，未组织的
[4] IDG（International Data Group）国际数据集团

increasing volume, variety[1] and velocity of information that forms big data.

1.1 Volume

Many factors contribute to the increase in data volume—transaction-based data stored through the years, text data constantly streaming in from social media, increasing amounts of sensor[2] data being collected, etc. In the past, excessive data volume created a storage issue. But with today's decreasing storage costs, other issues emerge, including how to determine relevance amidst the large volumes of data and how to create value from data that is relevant.

1.2 Variety

Data today comes in all types of formats—from traditional databases to hierarchical data stores created by end users and OLAP[3] systems, to text documents, E-mail, meter-collected data, video, audio, stock ticker data and financial transactions. By some estimates, 80 percent of an organization's data is not numeric! But it still must be included in analyses and decision making[4].

1.3 Velocity

According to Gartner, velocity "means both how fast data is being produced and how fast the data must be processed to meet demand". RFID tags and smart metering[5] are driving an increasing need to deal with torrents of data in near-real time. Reacting quickly enough to deal with velocity is a challenge to most organizations.

2. Big data according to SAS[6]

At SAS, we consider two other dimensions when thinking about big data.

2.1 Variability[7]

In addition to the increasing velocities and varieties of data, data flows[8] can be highly inconsistent[9] with periodic peaks. Is something big trending in the social media? Perhaps there is a high-profile IPO[10] looming. Maybe swimming with pigs in the Bahamas is suddenly the must-do vacation activity. Daily, seasonal and event-triggered peak data loads can be challenging to manage—especially with social media[11] involved.

---

[1] variety [vəˈraiəti] *n.* 品种，种类
[2] sensor [ˈsensə] *n.* 传感器
[3] OLAP (On – Line Analytical Processing) 联机分析处理
[4] decision making 决策，判定
[5] smart metering 智能仪表
[6] SAS (Statistical Analysis System) 统计分析系统，由美国北卡罗来纳州立大学 1966 年开发的统计分析软件
[7] variability [ˌvɛəriəˈbiliti] *n.* 可变性
[8] data flow 数据流
[9] inconsistent [ˌinkənˈsistənt] *adj.* 不一致的，不协调的，矛盾的
[10] IPO (Initial Public Offerings) 首次公开发行股票
[11] social media 社交媒体

2.2 Complexity

When you deal with huge volumes of data, it comes from multiple sources. It is quite an undertaking to link, match, cleanse and transform data across systems. However, it is necessary to connect and correlate[1] relationships, hierarchies and multiple data linkages or your data can quickly spiral out of control[2]. Data governance can help you determine how disparate[3] data relates to common definitions and how to systematically integrate[4] structured and unstructured data assets to produce high-quality information that is useful, appropriate and up-to-date.

Ultimately, regardless of the factors involved, we believe that the term big data is relative; it applies (per Gartner's assessment) whenever an organization's ability to handle, store and analyze data exceeds its current capacity.

3. Uses for big data

So the real issue is not that you are acquiring large amounts of data (because we are clearly already in the era of big data). It's what you do with your big data that matters. The hopeful vision for big data is that organizations will be able to harness relevant data and use it to make the best decisions.

Technologies today not only support the collection and storage of large amounts of data, they provide the ability to understand and take advantage of [5] its full value, which helps organizations run more efficiently and profitably. For instance, with big data and big data analytics, it is possible to:

(1) Analyze millions of SKUs[6] to determine optimal prices that maximize profit and clear inventory[7].

(2) Recalculate[8] entire risk portfolios in minutes and understand future possibilities to mitigate[9] risk.

(3) Mine customer data for insights that drive new strategies for customer acquisition, retention, campaign optimization and next best offers.

(4) Quickly identify customers who matter the most.

(5) Generate retail coupons[10] at the point of sale based on the customer's current and past purchases, ensuring a higher redemption rate[11].

---

[1] correlate [ˈkɔrileit] *vt.* 使相互关联
vi. 和……相关
[2] spiral out of control 失去控制，急剧上升
[3] disparate [ˈdispərit] *adj.* 全异的
[4] systematically integrate 系统集成
[5] take advantage of 利用
[6] SKU (Stock Keeping Unit) 库存量单位，表示最小存货单位
[7] inventory [ˈinvəntri] *n.* 库存，存货
[8] recalculate [riˈkælkjuleit] *vt.* 重新计算
[9] mitigate [ˈmitigeit] *v.* 减轻
[10] coupon [ˈkuːpɔn] *n.* 商家的优待券
[11] redemption rate 兑换率

(6) Send tailored recommendations to mobile devices at just the right time, while customers are in the right location to take advantage of offers.

(7) Analyze data from social media to detect new market trends and changes in demand.

(8) Use clickstream analysis and data mining to detect fraudulent behavior.

(9) Determine root causes of failures, issues and defects by investigating[1] user sessions, network logs and machine sensors.

Examples of big data:

(1) RFID (radio frequency ID) systems generate up to 1000 times the data of conventional bar code systems.

(2) 10000 payment card transactions are made every second around the world.

(3) Walmart handles more than 1 million customer transactions an hour.

(4) 340 million tweets are sent per day. That's nearly 4000 tweets per second.

(5) Facebook has more than 901 million active users generating social interaction data.

(6) More than 5 billion people are calling, texting, tweeting and browsing websites on mobile phones.

4. Challenges

Many organizations are concerned that the amount of amassed[2] data is becoming so large that it is difficult to find the most valuable[3] pieces of information.

(1) What if your data volume gets so large and varied you don't know how to deal with it?

(2) Do you store all your data?

(3) Do you analyze it all?

(4) How can you find out which data points are really important?

(5) How can you use it to your best advantage?

Until recently, organizations have been limited to using subsets[4] of their data, or they were constrained to simplistic[5] analyses because the sheer volumes of data overwhelmed[6] their processing platforms. What is the point of collecting and storing terabytes of data if you can't analyze it in full context, or if you have to wait hours or days to get results? On the other hand, not all business questions are better answered by bigger data.

You now have two choices:

Incorporate massive[7] data volumes in analysis. If the answers you are seeking will be better provided by analyzing all of your data, go for it. The game-changing technologies that extract[8] true

---

[1] investigate [in'vestigeit] *v.* 调查，研究

[2] amass [ə'mæs] *vt.* 收集，积聚

[3] valuable ['væljuəbl] *adj.* 有价值的

[4] subset ['sʌbset] *n.* 子集

[5] simplistic [sim'plistik] *adj.* 过分简单化的

[6] overwhelm [əuvə'hwelm] *vt.* 淹没，压倒

[7] massive ['mæsiv] *adj.* 大量的，大块的

[8] extract [iks'trækt] *vt.* 提取

value from big data—all of it—are here today. One approach is to apply high-performance analytics to analyze the massive amounts of data using technologies such as grid computing[1], in-database processing and in-memory analytics.

Determine upfront which big data is relevant. Traditionally, the trend has been to store everything (some call it data hoarding[2]) and only when you query the data do you discover what is relevant. We now have the ability to apply analytics on the front end[3] to determine data relevance based on context. This analysis can be used to determine which data should be included in analytical processes and which can be placed in low-cost storage for later availability if needed.

5. Technologies

A number of recent technology advancements[4] are enabling organizations to make the most of big data and big data analytics:

(1) Cheap, abundant storage and server processing capacity.

(2) Faster processors.

(3) Affordable large-memory capabilities, such as Hadoop[5].

(4) New storage and processing technologies designed specifically for large data volumes, including unstructured data.

(5) Parallel processing[6], clustering, MPP[7], virtualization, large grid environments, high connectivity and high throughputs.

(6) Cloud computing and other flexible resource allocation arrangements.

Big data technologies not only support the ability to collect large amounts of data, they provide the ability to understand it and take advantage of its value. The goal of all organizations with access to large data collections should be to harness[8] the most relevant data and use it for optimized decision making.

It is very important to understand that not all of your data will be relevant or useful. But how can you find the data points that matter most? It is a problem that is widely acknowledged. "Most businesses have made slow progress in extracting value from big data. And some companies attempt to use traditional data management practices on big data, only to learn that the old rules no longer apply," says Dan Briody, in the 2011 Economist Intelligence Unit's publication,"Big Data: Harnessing a Game-Changing Asset."

---

[1] grid computing 网格计算
[2] hoarding [ˈhɔːdiŋ] *n.* 储藏，囤积
[3] front end 前端
[4] advancement [ədˈvɑːnsmənt] *n.* 进步
[5] Hadoop 一个分布式系统基础架构，由 Apache 基金会开发
[6] parallel processing 并行处理
[7] MPP (Massively Parallel Processor) 大量信息并行处理机
[8] harness [ˈhɑːnis] *vt.* 利用

参 考 译 文

云 计 算

云计算是对通过因特网提供托管服务的总称。这些服务大致分为三类：基础设施即服务（IaaS）、平台即服务（PaaS）和软件即服务（SaaS）。云计算名称的灵感来自于云的符号，在流程图和图表中该符号通常用来代表因特网。

与传统托管不同，云服务有三个明显特点。云计算是按需（通常按分钟或小时）销售的；它有一定的弹性——用户可以在给定的时间内得到他们所需要的服务；这些服务完全由提供者来管理（客户只需一台个人计算机并将其接入因特网）。由于虚拟化和分布式计算中的重大创新，以及高速因特网和疲软的经济因素，人们对云计算的兴趣越来越浓。

1. 云

云可以是私有云或公共云。公共云向因特网的任何人出售服务。（目前，亚马逊 Web 服务是最大的公共云提供商。）私有云是一个专有网络或数据中心，为数量有限的人提供托管服务。服务提供商使用公共云创建自己的私有云，就被称为虚拟私有云。无论是私有云还是公共云，云计算的目标都是提供对计算资源和 IT 服务方便而可扩展的访问。

1.1 公共云

公共云是一个基于标准的云计算模式。在该模式中，提供商通过因特网向公众提供服务资源（如应用程序和存储）。公共云服务可以是免费的，也可以按使用次数收费。

使用公共云服务的主要好处是：

- 因为硬件、应用和带宽成本都由提供商负责，所以实现起来简单而廉价。
- 可根据需要扩展。
- 因为按使用付费，不会浪费资源。

公共云的例子包括：亚马逊弹性计算云（EC2），IBM 的蓝云、Sun 云，谷歌 AppEngine 和 Windows Azure 服务平台。

1.2 私有云

私有云（又称为内部云或企业云）是一个营销术语，它有自己的计算架构，为数量有限的防火墙用户提供托管服务。

虚拟化和分布式计算的发展，让企业网络和数据中心管理员能够有效地满足其公司内“客户”的需求。

使用词汇“私有云”的营销媒体是为了吸引需要或想更多地控制他们自己的数据的组织，这些组织可以得到通过第三方托管的服务（如亚马逊的弹性计算云或简单存储服务）来更多地控制自己的数据。

1.3 社区云

社区云由特定社区中的几个组织分享基础结构，它们有共同关心的问题（安全性、适应性、权限等），不论它是由内部管理或由第三方管理，也不管托管在内部或外部。社区云的成本承担的用户人数比公共云少（但比私有云多），所以只节省了实现云计算的部分潜在成本。

1.4 混合云

混合云是指两个或两个以上的云（私有云、社区云或公共云）的合成，由独特的实体组成但合起来使用，其益处在于能提供多种部署模式。这种组合扩展了云服务部署的方式，使 IT 组织使用公共云计算资源来满足临时需求。这种能力使混合云能使用跨越多个云的云爆发。

云爆发是一个应用部署模型，在该模型中，应用运行在私有云或数据中心，当所需的计算能力增加时，就“爆发”到公共云。云爆发和混合云模式的一个主要优点是，组织只需要为其所需的额外计算资源付费。

云爆发可以使数据中心建立一个支持平均负载的内部 IT 基础设施，并在需求高峰时使用公共云或私有云的云资源。

通过利用“混合云”架构，企业和个人都能够结合当地即时可用资源获得容错度，而不依赖因特网。混合云架构既需要本地资源也需要异地（远程）基于服务器的云计算基础设施。

混合云缺乏内部应用程序的灵活性、安全性和确定性。混合云既提供了本地应用的灵活性，还提供了基于云服务的容错能力和可扩展性。

2. 什么是 XaaS（万事即服务）

XaaS 是一组术语，表示以下多个意义：“X 即服务”、“万事即服务”或“一切皆服务”。它指日益增加的通过因特网而不是本地或现场提供的各种服务。XaaS 是云计算的本质。

XaaS 最常见的例子是 SaaS、IaaS 和 PaaS。将这三项组合使用，有时也称为 SPI 模式（SaaS、PaaS 和 IaaS）。XaaS 的其他例子包括存储即服务（SaaS）、通信即服务（CaaS）、网络即服务（NaaS）和监控即服务（MaaS）。

2.1 IaaS

IaaS 是一个供应模型，其中一个组织把支撑其业务的设备外包出去，这些设备包括存储、硬件、服务器和网络组件。服务提供者拥有该设备并负责建设机房、运行和维护。通常，客户按照使用付费。

IaaS 的特性和组件包括：

- 实用计算服务和计费模式。
- 自动化管理任务。
- 动态升级。

- 桌面虚拟化。
- 基于策略的服务。
- 因特网连接。

IaaS 是云计算服务的三个主要类别之一。另外两个是 SaaS 和 PaaS。

基础设施即服务有时也被称为硬件即服务（HaaS）。

2.2 PaaS

PaaS 是一种通过因特网租用硬件、操作系统、存储和网络容量的方法。这种服务交付模式允许用户租用虚拟服务器以及运行现有的应用程序，或者开发并测试新的应用程序所需要的相关服务。

PaaS 来自 SaaS。SaaS 是通过因特网向客户提供托管软件应用的软件分发模式。对开发者而言，PaaS 有几个优点：有了 PaaS，可以频繁改变和升级操作系统；不同地域的开发团队可以一起开发软件项目；可以得到来自不同国家的众多资源服务；通过使用单一经销商的基础服务来降低首次和持续的成本，而不是维护多个硬件设施，因为多个厂商提供的多个硬件设备功能往往重叠或有不兼容问题；通过统一编程开发也可以让总体费用最小化。

另一方面，如果产品需要专有服务接口或开发语言时，PaaS 就有了“锁住”风险。另一个潜在的问题是，产品的灵活性可能不能满足那些要求变化很快的用户的需求。

2.3 SaaS

SaaS 是一种软件发布模式，在该模式中，应用程序由供应商或服务提供商托管，并通过网络（通常是因特网）向客户提供。

由于支持 Web 服务、成熟的面向服务架构（SOA）、新开发方法（如 Ajax）的基础技术受到人们的欢迎，SaaS 将成为一个越来越普遍的交付模式。同时，宽带服务越来越多地支持来自世界各地的用户访问。

SaaS 与 ASP（应用服务提供商）和按需计算的软件交付模式密切相关。IDC（国际数据公司）为 SaaS 提供了两种差异不大的交付模式。托管应用管理（托管 AM）模式类似于 ASP：提供商为客户提供商用软件，并把它安装在 Web 上。在按需模式的软件中，提供商让客户通过网络使用专门用于 SaaS 发布的一个副本应用程序。

SaaS 模式的好处包括：

- 易于管理。
- 自动更新和补丁管理。
- 兼容性（所有用户都具有相同的软件版本）。
- 同理，更容易合作。
- 全球可访问性。

软件分发的传统模式是，购买软件并安装在个人计算机上，有时也称软件为一个产品。

Lesson 10

Text

Network Device

1. Firewall

A firewall is a set of related programs located at a network gateway server. It protects the resources of a private network from users from other networks. The term also implies the security policy that is used with the programs. An enterprise with an intranet that allows its workers to have access to the wider Internet installs a firewall to prevent outsiders from accessing its own private data resources and for controlling what outside resources its own users have access to.

Basically, a firewall, working closely with a router program, examines each network packet to determine whether to forward it toward its destination. A firewall also includes or works with a proxy server that makes network requests on behalf of workstation users. A firewall is often installed in a specially designated computer separate from the rest of the network so that no incoming request can get directly at private network resources.

There are a number of firewall screening methods. A simple one is to screen requests to make sure they come from acceptable (previously identified) domain name and Internet Protocol addresses. For mobile users, firewalls allow remote access in to the private network by the use of secure logon procedures and authentication certificates.

A number of companies make firewall products. Features include logging and reporting, automatic alarms at given thresholds of attack, and a graphical user interface for controlling the firewall.

Computer security borrows this term from firefighting, where it originated. In firefighting, a firewall is a barrier established to prevent the spread of fire.

2. Gateway

A gateway is a network point that acts as an entrance to another network. On the Internet, a node or stopping point can be either a gateway node or a host (end-point) node. Both the computers of Internet users and the computers that serve pages to users are host nodes. The computers that control traffic within your company's network or at your local Internet service provider (ISP) are gateway nodes.

In the network for an enterprise, a computer server acting as a gateway node is often also acting

as a proxy server and a firewall server. A gateway is often associated with both a router, which knows where to direct a given packet of data that arrives at the gateway, and a switch, which furnishes the actual path in and out of the gateway for a given packet.

3. Router

In packet-switched networks such as the Internet, a router is a device or, in some cases, software in a computer that determines the next network point to which a packet should be forwarded toward its destination. The router is connected to at least two networks and decides which way to send each information packet based on its current understanding of the state of the networks it is connected to. A router is located at any gateway where one network meets another, including each point-of-presence on the Internet. A router is often included as part of a network switch.

A router may create or maintain a table of the available routes and their conditions and use this information along with distance and cost algorithms to determine the best route for a given packet. Typically, a packet may travel through a number of network points with routers before arriving at its destination. Routing is a function associated with the Network layer (layer-3) in the Open Systems Interconnection (OSI) model. A layer-3 switch is a switch that can perform routing functions.

An edge router is a router that interfaces with an asynchronous transfer mode (ATM) network. A brouter is a network bridge combined with a router.

4. Bridge

In telecommunication networks, a bridge is a product that connects a local area network (LAN) to another local area network that uses the same protocol (for example, Ethernet or Token Ring). You can envision a bridge as being a device that decides whether a message from you to someone else is going to the local area network in your building or to someone on the local area network in the building across the street. A bridge examines each message on a LAN, "passing" those known to be within the same LAN, and forwarding those known to be on the other interconnected LAN (or LANs).

In bridging networks, computer or node addresses have no specific relationship to location. For this reason, messages are sent out to every address on the network and accepted only by the intended destination node. Bridges learn which addresses are on which network and develop a learning table so that subsequent messages can be forwarded to the right network.

Bridging networks are generally always interconnected local area networks since broadcasting every message to all possible destinations would flood a larger network with unnecessary traffic. For this reason, router networks such as the Internet use a scheme that assigns addresses to nodes so that a message or packet can be forwarded only in one general direction rather than forwarded in all directions.

A bridge works at the data-link (physical network) level of a network, copying a data frame from one network to the next network along the communications path.

A bridge is sometimes combined with a router in a product called a brouter.

5. Hub

In general, a hub is the central part of a wheel where the spokes come together. The term is familiar to frequent fliers who travel through airport "hubs" to make connecting flights from one point to another. In data communications, a hub is a place of convergence where data arrives from one or more directions and is forwarded out in one or more other directions. A hub usually includes a switch of some kind. And a product that is called a "switch" could usually be considered a hub as well. The distinction seems to be that the hub is the place where data comes together and the switch is what determines how and where data is forwarded from the place where data comes together. Regarded in its switching aspects, a hub can also include a router.

(1) In describing network topologies, a hub topology consists of a backbone (main circuit) to which a number of outgoing lines can be attached ("dropped"), each providing one or more connection port for device to attach to. For Internet users not connected to a local area network, this is the general topology used by your access provider. Other common network topologies are the bus network and the ring network. (Either of these could possibly feed into a hub network, using a bridge.)

(2) As a network product, a hub may include a group of MODEM cards for dial-in users, a gateway card for connections to a local area network (for example, an Ethernet or a Token Ring), and a connection to a line.

6. Switch

In a telecommunications network, a switch is a device that channels incoming data from any of multiple input ports to the specific output port that will take the data toward its intended destination. In the traditional circuit-switched telephone network, one or more switches are used to set up a dedicated though temporary connection or circuit for an exchange between two or more parties. On an Ethernet local area network (LAN), a switch determines from the physical device (Media Access Control or MAC) address in each incoming message frame which output port to forward it to and out of. In a wide area packed-switched network such as the Internet, a switch determines from the IP address in each packet which output port to use for the next part of its trip to the intended destination.

In the Open Systems Interconnection (OSI) communications model, a switch performs the layer 2 or Data-Link layer function. That is, it simply looks at each packet or data unit and determines from a physical address (the "MAC address") which device a data unit is intended for and switches it out toward that device. However, in wide area networks such as the Internet, the destination address requires a look-up in a routing table by a device known as a router. Some newer switches also perform routing functions (layer 3 or the Network layer functions in OSI) and are sometimes called IP switches.

On larger networks, the trip from one switch point to another in the network is called a hop. The time a switch takes to figure out where to forward a data unit is called its latency. The price paid

for having the flexibility that switches provide in a network is this latency. Switches are found at the backbone and gateway levels of a network where one network connects with another and at the subnetwork level where data is being forwarded close to its destination or origin. The former are often known as core switches and the latter as desktop switches.

In the simplest networks, a switch is not required for messages that are sent and received within the network. For example, a local area network may be organized in a Token Ring or bus arrangement in which each possible destination inspects each message and reads any message with its address.

New Words

router	[ˈrautə]	*n.* 路由器
gateway	[ˈgeitwei]	*n.* 网关，门，通路
protect	[prəˈtekt]	*vt.* 保护，关税保护，投保
enterprise	[ˈentəpraiz]	*n.* 企业；公司
examine	[igˈzæmin]	*vt.* 检查，调查；研究，分析
screening	[ˈskriːniŋ]	*n.* 筛选，屏蔽
mobile	[ˈməubail]	*adj.* 可移动的，易变的，机动的
secure	[siˈkjuə]	*adj.* 安全的，可靠的
logon	[ˈləugən]	*v.* 登录上网
logging	[ˈlɔgiŋ]	*n.* 存入，联机；记录
alarm	[əˈlɑːm]	*n.* 警报，警告器
		vt. 恐吓，警告
given	[ˈgivn]	*adj.* 约定的；特定的；指定的
threshold	[ˈθreʃəuld]	*n.* 门槛；阈值；起点，开端
attack	[əˈtæk]	*n.* 进攻，攻击，侵袭
		v. 攻击
term	[təːm]	*n.* 术语
firefighting	[ˈfaiəfaitiŋ]	*n.* 消防
barrier	[ˈbæriə]	*n.* （阻碍通道的）障碍物，栅栏，屏障
furnish	[ˈfəːniʃ]	*v.* 供给
presence	[ˈprezns]	*n.* 存在，到场，出席
cost	[kɔst]	*n.* 成本，价钱，代价
brouter	[ˈbrautə]	*n.* 桥式路由器
bridge	[bridʒ]	*n.* 网桥
Ethernet	[ˈiːθənet]	*n.* 以太网
envision	[inˈviʒən]	*vt.* 想象，预见，展望
interconnect	[ˌintəkəˈnekt]	*vt.* 使互相连接

subsequent	[ˈsʌbsikwənt]	*adj*. 后来的，并发的
broadcast	[ˈbrɔːdkɑːst]	*n*. 广播
flood	[flʌd]	*n*. 洪水，水灾
		vt. 淹没
		vi. 被水淹，涌进
direction	[diˈrekʃən]	*n*. 收件人地址
frame	[freim]	*n*. 帧，框架，画面
hub	[hʌb]	*n*. （网络）集线器
convergence	[kənˈvəːdʒəns]	*n*. 集中，收敛
topology	[təˈpɔlədʒi]	*n*. 拓扑，布局
backbone	[ˈbækbəun]	*n*. 骨干，脊椎，中枢
dedicated	[ˈdedikeitid]	*adj*. 专用的
hop	[hɔp]	*v*. 中继段
latency	[ˈleitənsi]	*n*. 等待时间
subnetwork	[sʌbˈnetwəːk]	*n*. 子网
inspect	[inˈspekt]	*vt*. 检查，审查

Phrases

be located at	位于
gateway server	网关服务器
security policy	安全策略
protect from [against]	防止……遭受……；使……免于，保护……使不受
prevent … from	阻止；制止
have access to	有权使用
data resources	数据资源
outside resources	外部资源
router program	路由程序
on behalf of …	代表……
separate… from	分离，分开
mobile user	移动用户
remote access	远程访问
logon procedure	登录规程
authentication certificate	证书
at the threshold of	在……的开始
graphical user interface	图形用户界面
computer security	计算机安全
packet-switched network	包交换网络
in some cases	在某些情况下

network switch	网络转接
network point	网点
best route	最佳路由
arrive at	到达
layer-3 switch	第三层交换
edge router	边式路由器
network bridge	网桥
local area network (LAN)	局域网
Token Ring	令牌网
send out	发送
data-link level	数据链路层
bus network	总线网络
ring network	环形网络
dial-in	拨号
input port	输入端口
output port	输出端口
circuit-switched	线路交换的
Media Access Control (MAC)	媒体访问控制
packed-switched network	包交换网络
physical address	物理地址
wide area network (WAN)	广域网
look-up	查表
core switch	中心交换
desktop switch	桌面交换

Abbreviations

OSI (Open Systems Interconnection)	开放式系统互联参考模型
ATM (asynchronous transfer mode)	异步传输模式

Notes

[1] An enterprise with an intranet that allows its workers to have access to the wider Internet installs a firewall to prevent outsiders from accessing its own private data resources and for controlling what outside resources its own users have access to.

本句中，an enterprise 是主语，installs 是谓语，a firewall 是宾语，to prevent outsiders from accessing its own private data resources and for controlling what outside resources its own users have access to 是目的状语，修饰谓语 installs，with an intranet 和 that allows its workers to have access to the wider Internet 作定语，修饰和限定主语 an enterprise。

[2] A firewall is often installed in a specially designated computer separate from the rest of the network so that no incoming request can get directly at private network resources.

本句中，separate from the rest of the network 是形容词短语作定语，修饰和限定 computer，so that no incoming request can get directly at private network resources 是一个状语从句，作目的状语。

[3] A gateway is often associated with both a router, which knows where to direct a given packet of data that arrives at the gateway, and a switch, which furnishes the actual path in and out of the gateway for a given packet.

本句中，有两个 which 引导的非限定性定语从句，其中 which knows where to direct a given packet of data that arrives at the gateway 修饰和限定 a router，which furnishes the actual path in and out of the gateway for a given packet 修饰和限定 a switch。

[4] The router is connected to at least two networks and decides which way to send each information packet based on its current understanding of the state of the networks it is connected to.

本句中，based on 的意思是"根据"，it is connected to 是一个定语从句，修饰和限定 the networks，it 指代 the router。

[5] For this reason, router networks such as the Internet use a scheme that assigns addresses to nodes so that a message or packet can be forwarded only in one general direction rather than forwarded in all directions.

本句中，that assigns addresses to nodes so that a message or packet can be forwarded only in one general direction rather than forwarded in all directions 是一个定语从句，修饰和限定 a scheme。在该定语从句中，so that a message or packet can be forwarded only in one general direction rather than forwarded in all directions 是一个目的状语从句，修饰和限定 assigns。rather than 的意思是"而不是"。

[6] In a telecommunications network, a switch is a device that channels incoming data from any of multiple input ports to the specific output port that will take the data toward its intended destination.

本句中，that channels incoming data from any of multiple input ports to the specific output port that will take the data toward its intended destination 是一个定语从句，修饰和限定 a device。在该定语从句中，that will take the data toward its intended destination 是一个定语从句，修饰和限定 the specific output port。

Grammar

介　词

介词又称前置词，它是一种虚词，不能在句子中单独承担任何成分。但它在英语中却起着十分重要的作用，在计算机行业中使用范围十分广泛。特别是一些常用介词，其搭配能力非常强，可以用来表示种种不同的意义。因此，掌握好介词是学好英语的关键之一。

1. 介词的类型

就介词的结构而言，可以分为简单介词与复合介词两种。

1）简单介词

由一个单词组成的介词称为简单介词。例如：in、on、about、at、behind、for、against、by、over、with、from、like、through、under、to、around、near、before、after 等。

2）复合介词

由两个或两个以上单词组成的介词称为复合介词，它们已经成为一个固定的词组，其作用相当于一个介词。复合介词可有以下三种构成：

（1） 副词＋介词。

例如：out of、up to、together with、apart from 等。

（2） 动词、形容词或连词＋介词。

例如：except for、due to、because of、but for、owing to 等。

（3） 介词＋名词＋介词。

例如：in front of、in spite of、on account of、with regard to、by means of、on behalf of、in accordance with、in addition to 等。

2. 介词的宾语

介词不能在句子中独立担任一个成分，而需要和一个名词或相当于名词的部分构成介词短语，共同在句子中充当一个成分。与介词一起构成介词短语的那个部分称为介词的宾语。介词的宾语主要由以下词来充当。

1）名词

请看下例：

【例】What is the manager going to talk about at the meeting?

经理打算在会上谈些什么？

【例】The professor will give us a talk on the development of computers in the past few years next Saturday.

下周六，教授将给我们作一个关于最近几年来计算机发展的报告。

【例】With the help of the professor, they finished the experiment.

在教授的帮助下，他们完成了那个实验。

2） 代词

请看下例：

【例】He had no money with him, so he didn't buy that book.

他身上没有带钱，因此就没有买那本书。

【例】You should look after yourself since you've grown up.

既然你已长大成人，就应该照顾好自己。

3）动名词短语

请看下例：

【例】Before operating the scanner, you should read this manual first.

在操作扫描仪之前，你应该首先阅读这个手册。

句中，operating the scanner 是一个动名词短语，作介词 Before 的宾语。

【例】With many software products, you must press the NumLock key before using the numeric keypad to type numbers.

对于许多软件产品而言，必须按下 NumLock 键才能使用小键盘输入数字。

句中，many software products 作介词 With 的宾语，using the numeric keypad to type numbers 是动名词短语，作介词 before 的宾语。

4）动词不定式短语

请看下例：

【例】He had no choice but to accept their challenge.

他别无选择，只好接受他们的挑战。

【例】She did nothing but complain.

她除了抱怨，什么也不干。

注意：不定式短语只可在少数几个介词之后作宾语。介词之后的不定式符号有时可以省略。由连接代词或副词引起的不定式短语也可在句子中作介词的宾语。请看下例：

【例】The old professor gave them some advice on how to learn computer well.

那位老教授就如何学好计算机给他们提了一些建议。

【例】He will give us a lesson on how to improve our oral English.

他将给我们上一堂如何提高英语口语的课。

5）名词性从句

请看下例：

【例】He is not at all interested in what the others are saying.

他一点都不在意别人说什么。

【例】I've no idea as to how much this printer costs.

我不知道这台打印机的价格。

【例】He is a very good worker except that he is somewhat bad-tempered.

他是个好工人，只是脾气不太好。

6）副词

请看下例：

【例】This laser printer was bought last year, since then it has been working very well.

这台激光打印机是去年买的。从那时起，它一直运行得很好。

【例】Their manager has just come back from abroad.

他们的经理刚从国外回来。

【例】Before long, Tom decided to buy a computer, too.

不久前，汤姆也决定买一台计算机。

7）形容词

请看下例：

【例】At first he didn't like this book, but now he does.

起初他不喜欢这本书，但他现在喜欢了。

【例】This machine is far from perfect.

这台机器远不是完美无瑕的。

【例】All their work was in vain.

他们的一切工作都劳而无功。

8）复合结构

请看下例：

【例】You may depend on her to post the letter for you.

你可以相信她会替你把那封信寄出去。

【例】We'll be looking forward to Tom's coming.

我们将期待着汤姆的到来。

【例】The manager was annoyed at your saying that.

听到你那样说，经理很生气。

注意：在口语中，介词之后的动名词的逻辑主语可用主格。因此，上两例句中的 Tom's 和 your 可换成 Tom 和 you。

9）介词短语

请看下例：

【例】They worked till after midnight.

他们一直工作到午夜之后。

【例】The old man looked at her from over his glasses.

那位老人从眼镜上方看着她。

【例】You can only find that book nowhere but in this bookstore.

除了在这家书店，在别的书店是找不到这本书的。

3. 介词短语在句子中的作用

1）作状语

（1）时间状语。请看下例：

【例】We've worked in that company for two years.

我们已经在那个公司工作两年了。

本句中，for two years 作时间状语。

【例】Who knows what will happen in the future?

谁能预卜未来?

（2）作地点状语。请看下例：

【例】We bought this plotter in Beijing.

我们在北京买的绘图仪。

【例】They live near the post office.

他们住在邮局附近。

（3）作目的状语。请看下例：

【例】He came here for his book.

他到这里来拿他的书。

【例】What are you doing that for?

你为什么要做这件事?

（4）作结果状语。请看下例：

【例】He worked himself to death.

他操劳致死。

（5）作原因状语。请看下例：

【例】Thank you for helping me.

谢谢你的帮助。

（6）作方式状语。请看下例：

【例】All the machines are controlled by a computer.

所有机器都由计算机控制。

【例】How did you come here? On foot.

你怎么到这来的? 步行。

（7）作条件状语。请看下例：

【例】They did it without my knowledge and without my consent.

他们做这件事我既不知道也没同意。

【例】We couldn't finish this design without his help.

没有他的帮助，我们就不能完成这项设计。

（8）作让步状语、伴随状语，表示范围及其他。

【例】With all your faults, I still like you.

尽管你有缺点，我仍然喜欢你。

【例】He lay on his bed with his hands under his head, thinking.

他躺在床上，手放在头下面，思考着。

【例】Computers are being widely used in many fields.

计算机正广泛地使用在许多领域。

【例】Please tell me what happened in detail.

请详细告诉我发生的事情。

【例】Our manager is not interested in doing business with that firm.

我们经理不喜欢跟那家公司做生意。

【例】On the whole, Jack has been a capable operator.

总的来说，杰克一向是个能干的操作员。

2）作定语

介词短语作定语时，通常放在它所修饰的词的后面。请看下例：

【例】The man in the shop told me where to buy this printer.

商店里的那个人告诉我到哪里去买打印机。

【例】Most of the products on display are computers.

展出的大多数产品是计算机。

【例】What is the advantage of reading this book?

读这本书有什么用呢?

【例】The students did the experiments in physics last week.

上个星期学生们做了物理实验。

3）作表语

请看下例：

【例】Computers are of various types.

计算机有各种不同的型号。

【例】This book is of great help to us all.

这本书对我们大家很有帮助。

【例】This kind of software is out of date.

这种软件已经过时了。

4）作宾语补足语

请看下例：

【例】They find this kind of machine out of date.

他们发现这种机器已经过时了。

【例】We consider this experiment of great importance.

我们认为这个实验很重要。

【例】They must keep the computer in good order.

他们必须使这台计算机正常工作。

【例】The secretary always keeps everything in good order.

秘书总把一切都安排得井井有条。

5）作主语补足语

请看下例：

【例】This book is found of great value.

人们发现这本书很有价值。

【例】These printers should be kept in repair.

这些打印机要经常维修。

4. 有关介词的比较

1）表示时间的介词

（1） in、at、on 表示时间时，都是“在”的意思，但它们所表示的含义不同，因此与之连用的词也不同。

- in 可以表示某一段时间，也可以表示在某一段时间内，它所指的时间可长可短。常与表示年、月、季、周、早上、下午、晚上的词连用。例如：in the morning、in the afternoon、in the evening、in a week、in 1995、in September、in the near future 等。
- on 表示特定的日期，常与表示日期、星期的词连用。例如：on Friday、on January the first。

  若表示上午、下午、晚上的词之前有一个修饰词时，也要使用 on 而不能使用 in。例如：on a cold winter evening、on Monday morning。
- at 表示某一时间过程中的某一点，常与表示钟点、时刻的词连用。例如：at seven o'clock、at that time、at present 等。在夜间也要用介词 at，即 at night。

（2） from、since、for 都可以表示时间。通常 from 表示时间的起点，后面常与 till 相呼应；since 表示从某一时刻起一直延续到另一时刻，since 常用在完成时的句子中；而 for 则表示一段时间，它既可用在完成时的句子中，也可用在其他时态的句子中。请看下例：

【例】In order to finish that difficult task in time, they worked from morning till night.

为了按时完成那项艰巨的任务，他们夜以继日地工作。

【例】He has worked in this factory since he graduated from college.

自从他大学毕业以来，就一直在这个工厂工作。

【例】Great changes have taken place in this field since 1990.

自 1990 年以来，这一领域发生了重大的变化。

【例】Dr. Smith has worked in this institute for ten years.

史密斯博士在这个研究所已经工作十年了。

【例】The manager decided to stay in America for two months.

经理决定在美国逗留两个月。

2）表示地点的介词

英语中表示地点的介词有：in、on、at、by、near、beneath、above、below、over、under、before、behind、between 等。其表示的位置不同：in 表示“在……里”；on 表示“在……上”；at 表示比较笼统的空间位置，不涉及与周围的人或物的相对关系，意思相当于汉语的“在……（地方）”；by 表示“在……旁边”；near 表示“在……附近”；beneath 表示“在……（正）下方”，可与 under 互换；above 表示一物处于比另一物更高的位置，它与 below 相对；over 表示一物处于一物的正上方，它与 under 相对；before 表示“在……的前面”，它与 behind 相对；between 表示“在……之间”。

3）表示方式的介词

英语中常用表示方式的介词有 by、with、in 等。

（1） by 表示通过或依靠某种手段、方法去做某一事，常与动名词连用。请看下例：

【例】Only by working hard can you learn English well.

只有努力学习，你才能学好英语。

【例】Move the selection box by pointing inside the box and dragging it to a new position.

将鼠标指向选择盒内并把它拖动到一个新位置，则可移动选择盒。

【例】The various idiomatic uses can only be learned by experience.

多种多样的习惯用法只能通过实际应用才能学会。

（2） with 表示使用某种工具，以某种方式、方法进行某种工作。请看下例：

【例】People measure voltage with a multimeter.

人们用万用表测量电压。

【例】You can solve this problem with PCTOOLS.

你可以用 PCTOOLS 解决这个问题。

【例】They solved that problem with great difficulty.

他们花了很大的劲才解决了这个问题。

（3） in 表示完成某一事物所用的方式、单位等。请看下例：

【例】Computer memory is measured in kilobytes or megabytes of information.

计算机的内存以信息的千字节或兆字节来计量。

【例】This law may be expressed in symbolic form.

这个定律可以用符号表示。

【例】Can you write in German?

你能用德语写吗?

【例】Please fill in this form in ink, not in pencil.

请用钢笔填写这张表格，不要使用铅笔。

Exercises

一、根据课文内容，判断以下叙述的正误。

（1） An enterprise with an intranet installs a firewall to prevent outsiders from accessing its own private data resources.

（2） The computers that control traffic within your company's network or at your local Internet service provider（ISP）are host nodes.

（3） In telecommunication networks, a bridge is a product that connects one local area network to another which can use different protocols.

（4） The job of a bridge is to examine each message on a LAN, "pass" those known to be within the same LAN, and forward those known to be on the other interconnected LAN（or LANs）.

（5） Since computer or node addresses have no specific relationship to location in bridging networks, messages are sent out to every address on the network and accepted only by the

intended destination node.

(6) A brouter is a product which combines a. bridge with a router.

(7) The switch is the place where data comes together and the hub is what determines how and where data is forwarded from the place where data comes together.

(8) In the traditional circuit-switched telephone network, only one switch is used to set up a dedicated though temporary connection or circuit for an exchange between two or more parties.

(9) On larger networks, the trip from one switch point to another in the network is called a hop.

(10) The time a switch takes to figure out where to forward a data unit is called its latency.

二、根据课文内容填空。

(1) A firewall is ______________________________.

(2) A firewall is often installed ____________________ separate from the rest of the network so that ______________ can get directly at private network resources.

(3) A gateway is ______________________________.

(4) Host nodes are ______________________________. And gateway nodes are ______________________________.

(5) A router is a device or, in some cases, software in a computer ______________________________.

(6) A router is located at ____________________ where ____________________, including each point-of-presence on the Internet.

(7) An edge router is a router that ______________________________.

(8) A brouter is ______________________________.

(9) In data communications, a hub is a place of convergence where data ______________ and ____________________.

(10) In a telecommunications network, a switch is a device ____________________ from ______________ to ______________________________.

三、选择与以下各条叙述意义最接近的词汇。

(1) The type of computer processing where the user of the system communicates directly with the system to input data and instructions and receive output.

(2) The boundary between two systems; a shared boundary between two systems.

(3) The capability of have two or more jobs in the computer at the same time. Execution of the program is interleaved so that in a time interval each job will have been (partly) processed. Processing is not simultaneous.

(4) A pictorial representation of processes and procedures for operation on data. A diagram that describes documents, processes, processes, and equipment used in processing data in a specific application.

(5) Performing tests and checks on input to ensure that the input operation is legal and that the input itself is correct. Pertaining to a wide variety of tests that can be applied to ensure the

correctness of data being input to a computer system.

供选择的答案：

A. decision table

B. environment

C. flowchart

D. input/output system

E. input validation

F. integrated circuit

G. interactive computing

H. interface

I. multiprogramming

四、从供选择的词汇中选择最合适的填在文中相应数字处。

With the widespread use of the personal computer, many authorities in the field of ____1____ have point out need for computer literacy. Unfortunately, there is no ____2____ agreement as to what term "computer literacy" means. Some feel that computer literacy means knowing how to make the computer "compute"; that is, knowing how to program computers in one or more programming languages.

Others feel that knowing how to program is merely a small segment of computer literacy. These people ____3____ the major emphasis in schools should be on teaching how to effectively use the many software packages that available. Still others suggest that computer literacy education is not required. They suggest that computers are being so rapidly integrated into our society that using a computer will be as ____4____ as using a telephone or a video tape recorder, and that special education will not be necessary. ____5____ of one's definition of computer literacy, it is recognized by most that learning to use a computer is indeed an important skill in modern society.

供 1 选择的答案：

A. culture

B. science

C. education

D. industry

供 2 和 4 选择的答案：

A. equal

B. universal

C. different

D. difficult

E. common

F. big

供 3 选择的答案：

A. claim

B. deny

C. define

D. call

供5选择的答案：

A. Importance

B. Instead

C. Because

D. Regardless

五、听短文，在画线处填写所听到的单词或词组。

The Internet is a specific ____1____. It consists of a worldwide ____2____ of governmental, academic, public, and ____3____ networks based upon the networking ____4____ of the Internet Protocol Suite. It is the ____5____ of the Advanced Research Projects Agency Network (ARPANET) developed by DARPA of the U. S. Department of Defense. The Internet is also the ____6____ underlying the World Wide Web (WWW). The 'Internet' is most commonly ____7____ with a capital 'I' as a proper noun, for historical reasons and to distinguish it from other generic internetworks.

Participants in the Internet use a diverse array of methods of several hundred documented, and often ____8____, protocols compatible with the Internet Protocol Suite and an addressing system (IP Addresses) ____9____ by the Internet Assigned Numbers Authority and address registries. Service providers and large enterprises exchange information about the reachability of their address spaces through the Border Gateway Protocol (BGP), forming a redundent world-wide mesh of ____10____.

六、计算机软件水平考试真题自测（系统分析师级）：选择填空。

An Enterprise Resource Planning (ERP) is built upon a commercial ____A____ that promises the seamless ____B____ of all the information flowing through the company ____B____ financial, accounting, human resources, supply chain and customer information. In implementation, all ERP systems include several features. The system is installed on a typical database management system. It requires initial setup according to the organization's process, but it may be ____C____ according to the organization's unique process requirements through a tool set contained within the ERP applications. Using ERP, ____D____ can be prescribed to automate approval processes through established chains of command. One of the methods used to effect rapid implementation of the ERP system is to conduct concurrent ____E____ sessions.

供选择的答案：

A.	(1) operating system	(2) middleware	(3) software package	(4) management system
B.	(1) association	(2) connection	(3) combination	(4) integration
C.	(1) customized	(2) made	(3) manufactured	(4) produced
D.	(1) functions	(2) processes	(3) information	(4) workflows
E.	(1) JRP	(2) BPR	(3) RAD	(4) JAD

Skill Training

计算机英文论文结语的写作

1. 结语的写作要求

结语是对文章的创造性、指导性、经验性的总结，它以自身的条理性、明确性、客观性反映了论文或研究成果的价值。结论与引言相呼应，同摘要一样可为读者和二次文献作者提供依据。结论不是对研究结果的简单重复，而是对研究结果的提升，主要包括：

（1）本研究结果说明了什么问题，得出了什么规律性的东西，解决了什么理论或实际问题。

作者应该像评审人一样评审自己工作的质量、价值、理论和实际意义。

结语应该比摘要更深入。摘要只是简述研究的主要内容和主要贡献，结语要说明研究的意义与贡献。因此，不能简单重复摘要的相关内容。

对论文创新内容的概括，措辞要准确、严谨，不能模棱两可，含糊其辞，不能使用“大概”、“也许”、“可能是”这类词。

可以具体包括以下内容：理论分析或实验的简明概况、结论性数据以及数据和论据的意义。可以总结为：结论与基本原理的联系与推广、结论及其解释与前人工作的比较、研究的理论意义与可能产生的实际应用，或应用前景。

（2）对前人有关问题的看法进行了哪些检验，哪些与本研究结果一致，哪些与本研究结果不一致，作者进行了哪些修正、补充、发展或否定。

（3）本研究的不足之处或遗留问题，是否存在例外情况或本论文尚难以解释或解决的问题，也可提出一些需要进一步研究本课题的建议。

上述要点（1）是必须的，而（2）和（3）则视论文的具体内容而定（可以有，也可以没有）；如果不能导出结论，也可以没有结论而进行必要的讨论。

结论段具有相对的独立性，应提供明确、具体的定性和定量的信息。对要点的表述应具体，不能用抽象、笼统的语言。行文要简短，不要再展开论述，不应对论文中各段的小结作简单重复。

研究成果或论文的真正价值是通过具体“结论”来体现的，所以结论段也不宜用如“本研究具有国际先进水平”、“本研究结果属国内首创”以及“本研究结果填补了国内空白”一类语句来做自我评价。

2. 常用句型

（1）关于结果的意义：

The results presented in this paper are (seem) …

The findings reported here is (quite) striking (remarkable, fascinating).

These preliminary finings are very reliable (encouraging, promising, convincing).

The results reported here prove (confirm, support, bear out) the hypothesis (assumption, observation) that…

The results shed (throw) some (new) light on behavior (nature, role) of…

The above findings can be viewed (approached) as follows (in terms of…, from another standpoint).

This fruitful work gives explanation to…

(2) 关于导致的结论:

Our findings suggest that …

These findings lead the author to a conclusion that …

Our data leave open the question of whether …

In the future, we will extend the present studies to …

Our work has contributed to the understanding of …

The research work has brought about a discovery of …

Reading Material

Common Uses of Internet

1. E-mail

The concept of sending electronic text messages between parties in a way analogous to mailing letters or memos predates the creation of the Internet. Even today it can be important to distinguish[1] between Internet and internal E-mail systems. Internet E-mail may travel and be stored unencrypted[2] on many other networks and machines out of both the sender's and the recipient's control. During this time it is quite possible for the content to be read and even tampered[3] with by third parties, if anyone considers it important enough. Purely internal or intranet mail systems, where the information never leaves the corporate or organization's network, are much more secure, although in any organization there will be IT[4] and other personnel whose job may involve monitoring and occasionally accessing the E-mail of other employees not addressed to them.

2. The World Wide Web

Many people use the terms Internet and World Wide Web (or just the Web[5]) interchangeably, but the two terms are not synonymous[6].

The World Wide Web is a huge set of interlinked[7] documents, images and other resources, linked by hyperlinks and URLs. These hyperlinks and URLs[8] allow the web servers and other

[1] distinguish [disˈtiŋgwiʃ] *v.* 区别，辨别
[2] unencrypt [ˌʌninˈkript] *v.* 未加密
[3] tamper [ˈtæmpə] *vi.* 修改，篡改
[4] IT (Information Technology) 信息技术
[5] Web [web] *n.* 万维网
[6] synonymous [siˈnɔniməs] *adj.* 同义的
[7] interlink [ˌintəˈliŋk] *n.* 连环，结合，互连
[8] URL (Uniform Resource Locator) 统一资源定位符

machines that store originals, and cached copies, of these resources to deliver them as required using HTTP (Hypertext Transfer Protocol)[1]. HTTP is only one of the communication protocols used on the Internet.

Web services also use HTTP to allow software systems to communicate in order to share and exchange business logic and data.

Software products that can access the resources of the Web are correctly termed user agents[2]. In normal use, web browsers, such as Internet Explorer and Firefox, access web pages and allow users to navigate from one to another via hyperlinks. Web documents may contain almost any combination of computer data including graphics, sounds, text, video, multimedia and interactive content including games, office applications and scientific demonstrations.

Many individuals and some companies and groups use "web logs" or blogs[3], which are largely used as easily updatable online diaries. Some commercial organisations encourage[4] staff to fill them with advice on their areas of specialization in the hope that visitors will be impressed by the expert knowledge and free information, and be attracted to the corporation as a result.

3. Remote access

The Internet allows computer users to connect to other computers and information stores easily wherever they may be across the world. They may do this with or without the use of security, authentication[5] and encryption technologies, depending on the requirements.

This is encouraging new ways of working from home, collaboration and information sharing in many industries. An accountant sitting at home can audit the books of a company based in another country, on a server situated in a third country that is remotely maintained by IT specialists[6] in a fourth. These accounts could have been created by home-working bookkeepers, in other remote locations, based on information E-mailed to them from offices all over the world. Some of these things were possible before the widespread[7] use of the Internet, but the cost of private leased lines would have made many of them infeasible[8] in practice.

An office worker away from his desk, perhaps on the other side of the world on a business trip or a holiday, can open a remote desktop session into his normal office PC using a secure Virtual Private Network (VPN)[9] connection via the Internet. This gives the worker complete access to all of his or her normal files and data, including E-mail and other applications, while away from the office.

---

[1] HTTP (Hypertext Transfer Protocol) 超文本传输协议
[2] agent [ˈeidʒənt] *n.* 代理
[3] blog [blɔg] *n.* 博客
[4] encourage [inˈkʌridʒ] *vt.* 鼓励
[5] authentication [ɔːˌθentiˈkeiʃən] *n.* 证书
[6] specialist [ˈspeʃəlist] *n.* 专家，行家
[7] widespread [ˈwaidspred] *adj.* 分布广泛的，普遍的
[5] infeasible [inˈfiːzəbl] *adj.* 不可实行的
[9] VPN (Virtual Private Network) 虚拟私有网

This concept is also referred to by some network security people as the Virtual Private Nightmare[1], because it extends the secure perimeter of a corporate network into its employees' homes; this has been the source of some notable security breaches, but also provides security for the workers.

4. Collaboration

The low cost and nearly instantaneous[2] sharing of ideas, knowledge, and skills has made collaborative work dramatically easier. Not only can a group cheaply communicate and test, but the wide reach of the Internet allows such groups to easily form in the first place, even among niche interests.

Internet "chat", whether in the form of IRC[3] "chat rooms" or channels, or via instant[4] messaging systems, allow colleagues to stay in touch in a very convenient way when working at their computers during the day. Messages can be sent and viewed even more quickly and conveniently than via E-mail. Extension to these systems may allow files to be exchanged, "whiteboard" drawings to be shared as well as voice and video contact between team members.

Version control systems allow collaborating teams to work on shared sets of documents without either accidentally[5] overwriting each other's work or having members wait until they get "sent" documents to be able to add their thoughts and changes.

5. File sharing

A computer file can be E-mailed to customers, colleagues and friends as an attachment. It can be uploaded to a website or FTP[6] server for easy download by others. It can be put into a "shared location" or onto a file server for instant use by colleagues. The load of bulk downloads to many users can be eased by the use of "mirror[7]" servers or peer-to-peer[8] networks.

These simple features of the Internet, over a worldwide basis, are changing the basis for the production, sale, and distribution[9] of anything that can be reduced to a computer file for transmission. This includes all manner of print publications, software products, news, music, film, video, photography, graphics and the other arts. This in turn has caused seismic shifts in each of the existing industries that previously controlled the production and distribution of these products.

Internet collaboration technology enables business and project teams to share documents, calendars and other information. Such collaboration occurs in a wide variety of areas including

---

[1] nightmare [ˈnaitmεə] *n.* 梦魇，噩梦

[2] instantaneous [ˌinstənˈteinjəs] *adj.* 瞬间的，即刻的，即时的

[3] IRC（Internet Relay Chatting）因特网在线聊天系统

[4] instant [ˈinstənt] *adj.* 即时的

[5] accidentally [æksiˈdentəli] *adv.* 偶然地，意外地

[6] FTP（File Transfer Protocol）文件传输协议

[7] mirror [ˈmirə] *n.* 镜像 *vt.* 映射

[8] peer-to-peer 点对点

[9] distribution [distriˈbjuːʃən] *n.* 分布，发布

scientific research, software development, conference planning, political activism and creative writing.

6. Streaming media

Many existing radio and television broadcasters provide Internet "feeds" of their live audio and video streams. They may also allow time-shift viewing or listening such as Preview, Classic Clips and Listen Again features. These providers have been joined by a range of pure Internet "broadcasters" who never had on-air licenses. This means that an Internet-connected device, such as a computer or something more specific, can be used to access on-line media in much the same way as was previously possible only with a television or radio receiver. The range of material is much wider. Podcasting is a variation on this theme, where, usually audio, material is first downloaded in full and then may be played back on a computer or shifted to a digital audio player[1] to be listened to on the move. These techniques using simple equipment allow anybody, with little censorship or licensing control, to broadcast audio-visual material on a worldwide basis.

Webcams[2] can be seen as an even lower-budget extension of this phenomenon[3]. While some webcams can give full-frame-rate video, the picture is usually either small or updates slowly. Video chat rooms, video conferencing, and remote controllable webcams are also popular. Many uses can be found for personal webcams in and around the home, with and without two-way[4] sound.

YouTube, sometimes described as an Internet phenomenon because of the vast amount of users and how rapidly the site's popularity has grown, was founded on February 15, 2005. It is now the leading website for free streaming video. It uses a flash-based web player which streams video files in the format FLV. Users are able to watch videos without signing up; however, if users do sign up[5] they are able to upload an unlimited amount of videos and they are given their own personal profile. It is currently estimated that there are 64,000,000 videos on YouTube, and it is also currently estimated that 825,000 new videos are uploaded every day.

7. Voice telephony (VoIP)

VoIP[6] stands for Voice over IP, where IP refers to the Internet Protocol that underlies all Internet communication. This phenomenon began as an optional two-way voice extension to some of the instant messaging systems that took off around the year 2000. In recent years many VoIP systems have become as easy to use and as convenient as a normal telephone. The benefit is that, as the Internet carries the actual voice traffic, VoIP can be free or cost much less than a normal telephone call, especially over long distances and especially for those with always-on Internet connections such as cable or ADSL[7].

---

[1] digital audio player 数字音频播放器
[2] webcam [ˈwebkæm] *n.* 网络摄像头
[3] phenomenon [fiˈnɔminən] *n.* 现象
[4] two-way 双向
[5] sign up 签约
[6] VoIP 通过 IP 传输语音
[7] ADSL (Asymmetrical Digital Subscriber Loop) 非对称数字用户环线

Thus, VoIP is maturing into a viable alternative to traditional telephones. Interoperability between different providers has improved and the ability to call or receive a call from a traditional telephone is available. Simple, inexpensive VoIP MODEMS are now available that eliminate the need for a PC.

Voice quality can still vary from call to call but is often equal to and can even exceed that of traditional calls.

Remaining problems for VoIP include emergency telephone number dialing and reliability. Currently, a few VoIP providers provide an emergency service, but it is not universally available. Traditional phones are line-powered and operate during a power failure; VoIP does not do so without a backup power source for the electronics.

Most VoIP providers offer unlimited national calling, but the direction in VoIP is clearly toward global coverage with unlimited minutes for a low monthly fee.

VoIP has also become increasingly popular within the gaming world, as a form of communication between players. Popular gaming VoIP clients include Ventrilo and Teamspeak, and there are others available also. The PlayStation 3 and Xbox 360 also offer VoIP chat features.

参 考 译 文

网 络 设 备

1. 防火墙

防火墙是网络网关服务器上的一组相关程序，它保护个人网络的资源不被来自其他网络的用户所用。这个术语也暗示使用程序的安全策略。一个有局域网并允许员工通过局域网访问广域网的企业，安装防火墙以阻止外部用户使用它自己的私有数据资源并控制自己的用户访问外部资源。

防火墙的主要工作是与路由器程序一起，检查每一个网络数据包以决定是否将其转发到目的地址。防火墙也包括代理服务器或与代理服务器一起工作，代理服务器代表工作站用户提出网络请求。防火墙经常安装在特别指定的、独立于网络的计算机上，这样就需要进入请求而不能直接得到个人网络资源。

有好几种防火墙筛选方法。一种简单的方法是筛选请求确保它们都来自可接受的（以前指定的）域名或因特网协议地址。对于移动用户来说，防火墙允许通过使用安全登录规程和鉴定证书来远程访问个人网络。

有些公司制造防火墙产品。功能包括在攻击开始的某些时候提供日志、报告和自动报警，并带有图形界面以便用户管理防火墙。

防火墙这个术语源于消防领域，计算机安全借用了它。在消防领域，防火墙是防止火势蔓延的一个屏障。

2. 网关

网关是进入其他网络的一个网点。在因特网中，一个节点或一个止点都可以是网关节点

或主（端点）节点。因特网用户使用的计算机和给用户提供页面的计算机都是主节点。控制公司网络中流量的计算机或当地因特网服务提供商的计算机都是网关节点。

在一个企业的网络中，作为网关节点的计算机服务器也经常作为代理服务器或防火墙服务器。网关也往往与路由器和交换机相关。路由器知道把到达网关的数据包引到何处，交换机为进出网关的数据包提供实际的路径。

3. 路由器

在像因特网这样的交换网络中，路由器是一个设备，有时也是计算机中的一个软件，它决定数据包要转发的下一个网络节点。路由器至少连接两个网络，并根据它对当前相联网络状况的了解来决定发送每一个数据包的路径。路由器位于任意网关（一个网络与另一个网络连接处），包括因特网的每一个现存点。路由器也是交换机的一部分。

路由器可以建立或维护一个可用的路由表及它们的状况表，并且根据这些信息、传输距离和成本来决定发送某一数据包的最佳路由，典型地说，一个数据包可能会通过带路由器的许多网络节点才能到达目的地。路由器的功能与开放式系统互联参考模型的网络层（第三层）相关。第三层交换可以执行路由功能。

边式路由器是接口带异步传输模式网络的路由器。桥式路由器是结合了路由器的网桥。

4. 网桥

在通信网络中，网桥是把局域网与其他使用相同协议（例如以太网和令牌网）的局域网连接起来的产品。可以把网桥想象为一个设备，该设备决定是否把来自你的消息发送给你所在办公楼中的某人或街对面大楼中的某人。网桥检查局域网中的每一个消息，在同一局域网中传递这些已知的消息，并将其转发给其他相连接的局域网。

在网桥连接的网络中，计算机或节点地址与位置无特定的关系。因此，消息发送给网络中的每一个地址，但只被目的节点接收。网桥了解网络地址，并作出一个“信息表”，以便后来的消息可以转发给正确的网络。

网桥连接的网络通常总是与局域网连接，因为把每一个消息都传播给全部可能的目的节点会使大型网络因为不必要的通信而阻塞。因此，路由器连接的网络（如因特网）使用给节点分配地址的方案，以便消息或数据包只可以转发给一个地址，而不是转发给所有地址。

网桥工作在数据链路层（物理网），沿着通信路径把数据帧从一个网络复制到另一个网络。

网桥有时也和路由器结合在一起，组成桥式路由器的产品。

5. 集线器

通常，集线器是一个带轮辐轮子的中心部件。经常坐飞机的人熟悉这个术语，他们经过航空“中心”，从一个地方飞到另一个地方。在数据通信中，集线器是一个中心，把来自一个或多个地址的数据转发到另外的一个或多个地址。集线器通常包括某种类型的交换机。被称为交换机的产品通常也被认为是一种集线器。区别在于集线器似乎是数据集中的地方，而交换机决定把数据从数据集中的地方如何转发出去以及发往何处。从其交换方面看，集线器也可以包括路由器。

（1）在描述网络拓扑结构时，集线器拓扑由一个主干网（主线路）构成，许多外线都可以连接（“下行”）到主干网，每个集线器都为连接设备提供一个或多个连接口。对于没有连接到一个局域网的因特网用户来说，这是你的接入提供商常用的拓扑结构。其他常用的网络拓扑结构有总线网和令牌网。（它们都可以通过网桥传入集线器网络。）

（2）作为一种网络产品，集线器也许会包括一组用于拨入用户的调制解调器卡，一个用于连接到局域网（如以太网和令牌网）的网关卡和一个线路连接口。

6. 交换机

在通信网络中，交换机是一个设备，它把来自多个输入口的数据引导到特定的输出口，该输出口把数据转发到目的地。在传统的线路交换电话网中，为了在两个或多个部分之间交换，要进行一个或多个交换来建立临时的专用通信连接或线路。在以太局域网中，交换机根据每个到达消息帧中的物理设备（媒体访问控制或 MAC）的地址决定用哪个输出口转发和输出。在像因特网这样的广域交换网络中，交换机根据每一个包中的 IP 地址决定下一步到达目的地使用哪个输出口。

在开放式系统互联通信模型中，交换机执行第二层或数据链路层功能，即它简单地查看每一个数据包或数据单元，根据其物理地址（MAC 地址）决定数据单元的目的设备并把它发送给该设备。然而，在像因特网这样的广域网中，目的地址需要查看有路由器提供的路由表。一些较新的交换机也执行路由功能（第三层或 OSI 的网络层功能），有时也称其为 IP 交换机。

在更大的网络中，从一个交换点到另一个交换点的旅行被称为“中继”。转发数据单元所用的交换时间称为“等待时间”。这个等待时间就是交换机在网络中提供灵活性的代价。交换机位于主干网和网关以及子网络中，在主干网和网关中，一个网络与另一个网络连接；在子网络中，数据几乎被转发到目的地或发回原处。前者通常被称为核心交换，后者为桌面交换。

在最简单的网络中，无须用交换机来发送和接收网内的消息。例如，可以用令牌网或布置总线的形式组织局域网，每一个可能的目的文件可以检查每一消息并且可以读取带有地址的任何消息。

Lesson 11

Text

Software Engineering

1. Software Engineering Basics

Software engineering is the process of manufacturing software systems. A software system consists of executable computer code and the supporting documents needed to manufacture, use, and maintain the code. For example, a word processing system consists of an executable program (the word processor), user manuals and the documents, such as requirements and designs, needed to produce the executable program and manuals.

Software engineering is ever more important as larger, more complex, and life-critical software systems proliferate. The rapid decline in the costs of computer hardware means that the software in a typical system often costs more than the hardware it runs on. Large software systems may be the most complex things ever built. This places great demands on the software engineering process, which must be disciplined and controlled.

The software engineering process itself is usually divided into phases. The definition of these phases, their ordering, and the interactions between the phases specify a software life-cycle model. The best-known life-cycle model is the waterfall model consisting of a requirements definition phase, a design phase, a coding phase, a testing phase and a maintenance phase. The output of each phase serves as the input to the next.

The purpose of the requirements phase is to define what a system should do and the constraints under which it must operate. This information is recorded in a requirements document. A typical requirements document might include a product overview; a specification of the development, operating and maintenance environment for the product; a high-level conceptual model of the system; a specification of the user interface; specification of functional requirements; specification of nonfunctional requirements; specification of interfaces to systems outside the system under development; specification of how errors will be handled; and a listing of possible changes and enhancements to the system. Each requirement, usually numbered for reference, must be testable.

In the design phase, a plan is developed for how the system will implement the requirements. The plan is expressed using a design method and notation. Many methods and notations for software design have been developed. Each method focuses on certain aspects of a system and ignores or minimizes others. This is similar to viewing a building with an architectural drawing, a plumbing

diagram, an electrical wiring diagram, and so forth.

The coding phase of the software life-cycle is concerned with the development of code that will implement the design. This code is written in a formal language called a programming language. Programming languages have evolved over time from sequences of ones and zeros directly interpretable by a computer, through symbolic machine code, assembly languages, and finally to higher-level languages that are more understandable to humans.

Most coding today is done in one of the higher-level languages. When code is written in a higher-level language, it is translated into assembly code, and eventually machine code, by a compiler. Many higher-level languages have been developed, and they can be categorized as functional languages, declarative languages and imperative languages.

Following the principle of modularity, code on large systems is separated into modules, and the modules are assigned to individual programmers. A programmer typically writes the code using a text editor. Sometimes a syntax-directed editor that "knows" about a given programming language and can provide programming templates and check code for syntax errors is used. Various other tools may be used by a programmer, including a debugger that helps find errors in the code, a profiler that shows which parts of a module spend most time executing, and optimizers that make the code run faster.

Testing is the process of examining a software product to find errors. This is necessary not just for code but for all life-cycle products and all documents in support of the software such as user manuals.

The software testing process is often divided into phases. The first phase is unit testing of software developed by a single programmer. The second phase is integration testing where units are combined and tested as a group. System testing is done on the entire system, usually with test cases developed from the system requirements. Acceptance testing of the system is done by its intended users.

The basic unit of testing is the test case. A test case consists of a test case type, which is the aspect of the system that the test case is supposed to exercise; test conditions, which consist of the input values for the test; the environmental state of the system to be used in the test; and the expected behavior of the system given the inputs and environmental factors.

When software is changed to fix a bug or add an enhancement, a serious error is often introduced. To ensure that this does not happen, all test cases must be rerun after each change. The process of rerunning test cases to ensure that no error has been introduced is called regression testing.

Walkthroughs and inspections are used to improve the quality of the software development process. Consequently, the software products created by the process are improved. A quality system is a collection of techniques whose application results in continuous improvement in the quality of the development process. Elements of the quality system include reviews, inspections and process audits.

Large software systems are not static; rather, they change frequently both during development

and after deployment. Maintenance is the phase of the software life-cycle after deployment. The maintenance phase may cost more than all of the others combined and is thus of primary concern to software organizations. The Y2K (Year 2000) problem was, for example, a maintenance problem.

Maintenance consists of three activities: adaptation, correction and enhancement. Enhancement is the process of adding new functionality to a system. This is usually done at the request of system users. This activity requires a full life-cycle of its own. That is, enhancements demand requirements, design, implementation and test. Studies have shown that about half of maintenance effort is spent on enhancements.

Adaptive maintenance is the process of changing a system to adapt it to a new operating environment, for example, moving a system from the Windows operating system to the Linux operating system. Adaptive maintenance has been found to account for about a quarter of total maintenance effort. Corrective maintenance is the process of fixing errors in a system after release. Corrective maintenance takes about 20% of maintenance effort.

Since software systems change frequently over time, an important activity is software configuration management. This consists of tracking versions of life-cycle objects, controlling changes to them, and monitoring relationships among them. Configuration management activities include version control, which involves keeping track of versions of life-cycle objects; change control, an orderly process of handling change requests to a system; and build control, the tracking of which versions of work products go together to form a given version of a software product.

2. Requirements Analysis

In software engineering, requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered device, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirements analysis is critical to the success of a development project.

Systematic requirements analysis is also known as requirements engineering. It is sometimes referred to loosely by names such as requirements gathering, requirements capture or requirements specification.

Requirements must be actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

3. Software Design

Software design is a process of problem-solving and planning for a software solution. After the purpose and specifications of software is determined, software developers will design or employ designers to develop a plan for a solution. It includes low-level component and algorithm implementation issues as well as the architectural view.

The software requirements analysis (SRA) step of a software development process yields specifications that are used in software engineering. If the software is "semiautomated" or user-centered, software design may involve user experience design yielding a storyboard to help determine

those specifications. If the software is completely automated (meaning no user or user interface), a software design may be as simple as a flow chart or text describing a planned sequence of events. There are also semi-formal methods like Unified Modeling Language and Fundamental modeling concepts. In either case some documentation of the plan is usually the product of the design.

A software design may be platform-independent or platform-specific, depending on the availability of the technology called for by the design.

There are many aspects to consider in the design of a piece of software. The importance of each should reflect the goals the software is trying to achieve. Some of these aspects are:

- Extensibility—New capabilities can be added to the software without major changes to the underlying architecture.
- Robustness—The software is able to operate under stress or tolerate unpredictable or invalid input. For example, it can be designed with resilience to low memory conditions.
- Reliability—The software is able to perform a required function under stated conditions for a specified period of time.
- Fault-tolerance—The software is resistant to and able to recover from component failure.
- Security—The software is able to withstand hostile acts and influences.
- Maintainability—The software can be restored to a specified condition within a specified period of time. For example, antivirus software may include the ability to periodically receive virus definition updates in order to maintain the software's effectiveness.
- Compatibility—The software is able to operate with other products that are designed for interoperability with another product. For example, a piece of software may be backward-compatible with an older version of itself.
- Modularity—The resulting software comprises of well defined, independent components. That leads to better maintainability. The components could be then implemented and tested in isolation before being integrated to form a desired software system. This allows division of work in a software development project.
- Reuse—the modular components designed should capture the essence of the functionality expected out of them and no more or less. This single-minded purpose render the components reusable wherever there are similar needs in other designs.

New Words

code	[kəud]	*n.* 代码，编码 *v.* 编码
maintain	[mein'tein]	*vt.* 维护，维修
manual	['mænjuəl]	*n.* 手册，指南 *adj.* 手的，手动的
proliferate	[prə'lifəreit]	*v.* 激增，扩散

decline	[diˈklain]	*vi.* 下倾，下降
disciplined	[ˈdisiplind]	*n.* 受过训练的，遵守纪律的
phase	[feiz]	*n.* 阶段，状态，相，相位
waterfall	[ˈwɔːtəfɔːl]	*n.* 瀑布
specification	[ˌspesifiˈkeiʃən]	*n.* 详述，规格，说明书，规范
handle	[ˈhændl]	*vt.* 处理，操作
		n. 句柄
notation	[nəuˈteiʃən]	*n.* 记号，符号
ignore	[igˈnɔː]	*vt.* 不理睬，忽视，忽略
minimize	[ˈminimaiz]	*v.* 最小化
interpretable	[inˈtəːprətəbl]	*adj.* 能说明的，能翻译的，可判断的
symbolic	[simˈbɔlik]	*adj.* 符号的
module	[ˈmɔdjuːl]	*n.* 模块
template	[ˈtemplit]	*n.* 模板（= templet）
optimize	[ˈɔptimaiz]	*vt.* 使最优化
behavior	[biˈheivjə]	*n.* 举止，行为
bug	[bʌg]	*n.* 故障，小虫
serious	[ˈsiəriəs]	*adj.* 严重的
walkthrough	[ˈwɔːkθruː]	*n.* 预排
inspection	[inˈspekʃən]	*n.* 检查，审查
deployment	[diˈplɔiment]	*n.* 部署
activity	[ækˈtiviti]	*n.* 行动，行为
adaptation	[ˌædæpˈteiʃən]	*n.* 适应，改编
functionality	[ˌfʌŋkəʃəˈnæliti]	*n.* 功能性，泛函性
release	[riˈliːs]	*n.* 版本，发布
encompass	[inˈkʌmpəs]	*vt.* 包括，包含
project	[ˈprɔdʒekt]	*n.* 项目，工程
gathering	[ˈgæðəriŋ]	*n.* 聚集，收款
capture	[ˈkæptʃə]	*n.* &*vt.* 捕捉
actionable	[ˈækʃənəbəl]	*adj.* 可行动的
semiautomated	[ˈsemiɔːtəˈmeitid]	*adj.* 半自动的
extensibility	[iksˌtensəˈbiliti]	*n.* 延伸性，伸展性
stress	[stres]	*n.* 重压，压力
unpredictable	[ˌʌnpriˈdiktəbl]	*adj.* 不可预知的
resilience	[riˈzisliəns]	*n.* 弹性
reliability	[riˌlaiəˈbiliti]	*n.* 可靠性
fault-tolerance	[fɔːlt-ˈtɔlərəns]	*n.* 容错
resistant	[riˈzistənt]	*adj.* 抵抗的

withstand	[wið'stænd]	*vt.* 抵挡，经受住
condition	[kən'diʃən]	*n.* 情况，状态
compatibility	[kəmˌpæti'biliti]	*n.* 兼容性
interoperability	['intərˌɔpərə'biliti]	*n.* 互通性，互用性，协同工作的能力
modularity	[ˌmɔdju'læriti]	*n.* 积木性，模式化，模块性
isolation	[ˌaisəu'leiʃən]	*n.* 隔绝，孤立，隔离
reuse	[riː'juːz]	*vt.* 再使用，复用
		n. 重新使用

Phrases

word processing system	文字处理系统，字处理系统
word processor	字处理器，字处理程序
user manual	用户手册
be divided into	被分为
software life-cycle model	软件生命周期模型
waterfall model	瀑布模型
requirement document	需求文档
functional requirement	功能需求
focus on	集中
assembly language	汇编语言
higher-level language	高级语言
text editor	文本编辑器
test case	测试案例，测试用例
acceptance testing	验收测试
test condition	测试条件
regression testing	回归测试
process audit	过程审核
adaptive maintenance	适应性维护
account for	说明；（在数量、比例上）占；解决，得分
corrective maintenance	更正性维护，纠正性维护
requirement analysis	需求分析
requirements engineering	需求工程
flow chart	流程图
Unified Modeling Language	统一建模语言
stated condition	给定条件

Abbreviations

Y2K（Year 2000）	千年虫

SRA（software requirements analysis）　　软件需求分析

Notes

[1] Various other tools may be used by a programmer, including a debugger that helps find errors in the code, a profiler that shows which parts of a module spend most time executing, and optimizers that make the code run faster.

本句中，有三个定语从句：that helps find errors in the code 修饰和限定 a debugger；that shows which parts of a module spend most time executing 修饰和限定 a profiler；that make the code run faster 修饰和限定 optimizers。a debugger、a profiler 和 optimizers 都是程序员可以使用的工具。

[2] A test case consists of a test case type, which is the aspect of the system that the test case is supposed to exercise; test conditions, which consist of the input values for the test; the environmental state of the system to be used in the test; and the expected behavior of the system given the inputs and environmental factors.

本句中，which is the aspect of the system that the test case is supposed to exercise 是一个非限定性定语从句，对 a test case type 进行补充说明。在该从句中，that the test case is supposed to exercise 是一个定语从句，修饰和限定 the aspect of the system。which consist of the input values for the test 也是一个非限定性定语从句，对 test conditions 进行补充说明。

[3] Configuration management activities include version control, which involves keeping track of versions of life-cycle objects; change control, an orderly process of handling change requests to a system; and build control, the tracking of which versions of work products go together to form a given version of a software product.

本句中，configuration management activities 作主语，include 作谓语，宾语是 version control，change control 和 build control。which involves keeping track of versions of life-cycle objects 是一个非限定性定语从句，对 version control 进行补充说明。在该从句中，involve doing sth. 的意思是“涉及/包括做某事”，keep track of 的意思是“跟踪”。an orderly process of handling change requests to a system 是一个名词性短语，对 change control 进行补充说明。在该短语中，handle 的意思是“处理”。the tracking of which versions of work products go together to form a given version of a software product 对 build control 进行补充说明。

Grammar

数　　词

英语中表示数目和顺序的词称为数词。数词主要分为基数词和序数词。若将基数词和序数词搭配，则可构成分数词。

1. 基数词

1）最基本的基数词

英语中表示数目多少的数词称为基数词。例如：one、two、six、eight 等。最基本的基数词是 1 到 20 和整十位数。1 到 12 的基数词是单独的单词，需单独记忆。而 13 到 19 的基数词则是在个位基数词后加后缀-teen，只是个别词在拼写上有变化（如：thirteen，eighteen，fifteen 等）。20 到 90 的整十位数词都有后缀-ty（如：sixty，forty 等）。

2）其他基数词的构成方法

（1）21 到 99：先写出“几十”，再写出“几”，两者之间用连字符连接。例如：

56　　fifty-six

98　　ninety-eight

（2）101 到 999：先写出“几百”，再写出“and”，最后写出末两位数或末位数。例如：

168　　one hundred and sixty-eight

888　　eight hundred and eighty-eight

503　　five hundred and three

（3）1000 以上的数：先从后往前数，每三位加一个“,”（逗号）。第一个“,”（逗号）前读作“thousand”；第二个“,”（逗号）前读作“million”；第三个“,”（逗号）前读作“billion”（美式英语）或 thousand million（英式英语）；然后一节一节表示。例如：

2,008	two thousand and eight
8,465	eight thousand, four hundred and sixty five
36,792	thirty-six thousand, seven hundred and ninety-two
579,283	five hundred and seventy-nine thousand, two hundred and eighty-two
18,521,002	eighteen million, five hundred and twenty-one thousand and two
205,000,000	two hundred and five million
9,000,000,000	nine billion（美式）；nine thousand million（英式）
6,000,000,000,000	six trillion（美式）；six billion（英式）

注意：若 hundred、thousand、million 等词前有数字时，它们一般为单数形式；若无数词时，可以复数形式出现，表示“数百”、“数千”、“数百万”。也可重叠使用，表示“成千上万”、“千千万万”等，这时，它们后面通常跟有介词 of + 名词。请看下例：

【例】Hundreds of new buildings have been built here in the recent years.

近年来，这里盖了数百座楼房。

【例】We all know the earth is millions of miles away from the sun.

我们都知道地球离太阳有数百万英里远。

【例】Hundreds of thousands of people come to visit this place every day.

每天都有成千上万的人参观这个地方。

【例】Thousands upon thousands of people are against this project.

千千万万的人都反对这项工程。

注意：若把表示“十”的数词变为复数，则可用来表示人的年龄或年代。请看下例：

【例】He became a computer expert in his thirties.

他三十多岁就成了一名计算机专家。

【例】It seems that their manager is in his twenties.

好像他们的经理有二十多岁。

【例】His first book was published in the early 1990s.

他的第一本书是在20世纪90年代初期出版的。

2. 序数词

1）序数词的构成

英语中表示顺序的数词称为序数词。一般情况下，序数词由其相应的基数词加词尾th构成，如：sixth。

（1）1到12基数词在变为序数词时，其拼写发生变化。例如：

one—first

two—second

three—third

five—fifth

eighth—eighth

nine—ninth

twelve—twelfth

（2）十以上的整十位数变成序数词时，变y为i，再加eth。例如：

twenty—twentieth

seventy—seventieth

（3）多位数词变为序数词时，只需将最后一个数词变为序数词。例如：

sixty-eight—sixty-eighth

seven hundred and forty-five—seven hundred and forty-fifth

注意：序数词有时用缩写形式。例如：

first—1st

second—2nd

third—3rd

ninth—9th

five—5th.

2）序数词的作用

序数词的主要作用是在句子中作定语，前面要加定冠词。请看下例：

【例】He lives on the third floor.

他住在三楼。

【例】The experiment was successful on the sixth time.

第六次实验成功了。

序数词前面也可以用不定冠词，其意思是“又一”、“再一”。请看下例：

【例】Would you please explain this sentence a second time?

你能把这个句子再解释一遍吗？

【例】He rose up a third time.

他又一次站了起来。

3. 分数词

1）分数

分数词由基数词和序数词搭配而成。分子用基数词，分母用序数词。当分子是1时，序数词用单数；若分子大于1时，序数词用复数形式。例如：

1/5	one-fifth
1/8	one-eighth
3/4	three-fourths 或 three quarters
1/4	one-fourth 或 one quarter
1/2	a half
$3\frac{2}{5}$	three and two-fifths

2）小数

英语中，把小数点（.）读作“point”。零读作“o”（字母“o”的音），也可读作 zero 或 nought。例如：

0.5	zero point five
0.03	zero point zero three
508.24	five o eight point two four

3）百分数

百分数用 per cent 表示，既可与 by 连用，也可单独使用。在句子中主要作状语。请看下例：

【例】The output of computers in China last year was 20 per cent more than in 2007.

去年中国计算机的产量比2007上升了20%。

【例】It was said the price of pin printers was reduced by 15 per cent.

据说针式打印机的价格下降了15%。

【例】Its total output value increased by 10 per cent over the previous year.

它的总产值比去年增长了10%。

4. 倍数

倍数的表达在计算机英语中时常出现，稍有不慎，就会产生理解错误。而倍数也是“汉译英”及“英译汉”中的难点。故在翻译倍数时，应仔细推敲。

1）简单表示法

在简单的倍数表示法中，主要使用… ×times as…as、… × times the size 或 length、width、height、depth amount of…以及… ×times + ×比较级 + than…。请看下例：

【例】This city is three times as large as that one.

这座城市是那座城市（面积）的三倍。

【例】The speed of this new printer is four times faster than that old one.

这台新打印机的速度比那台旧打印机快四倍。

【例】Their factory is five times the size of ours.

他们厂的规模是我们的五倍。

2）增减表示法

增、减表示法是一种较为复杂的表示法。

（1）增加类。

在 increase 等表示“增加”或“提高”词的后面也可用 times。若是“increase by × times”则翻译为“增加了×倍”或“增加到原来的× +1 倍”。请看下例：

【例】The population of this city has increased by four times as compared with that of 2000.

与 2000 年相比，该城市的人口已增加了四倍。

【例】Next year the output of computers will increase by five times.

明年计算机的产量将增加五倍。

注意，increase by 后面除可用“ × times”外，还可以使用其他数词。increase 后面也可跟介词 to，表示“增加到…”。请看下例：

【例】The number of the people who have home computers has increased to 20 per cent.

拥有家用计算机的人数已增加到 20%。

【例】This year the output of laser printers in their factory has increased to 15,000.

今年他们厂激光打印机的产量已经增加到 15,000 台。

（2）减少类。

在“decrease”或“reduce”等表示“减少”或“降低”的词的后面也常用 times。若是“decrease by × times”，常译为“减少到 1/×”或“减少了 1 -1/×”。请看下例：

【例】The weight of the new device has decreased by four times.

这种新设备的质量减少了四分之三。

【例】The new equipment they are designing will reduce error probability by three times and its speed will increase by four times.

他们正在设计的那种新设备将使误差率降低到三分之一，而速度将增加四倍。

注意：decrease by 也可使用其他数词。decrease to 表示“减少到……”。请看下例：

【例】The price of this type of computer has decreased by 30 per cent compared with that of last year.

这种计算机的价格比去年减少了 30%。

【例】The output of letter-quality printers has decreased to 3000 this year.

今年字符打印机的产量已下降到 3000 台。

注意：by 后面表示的是净增减数，by 有时可以省略，而 to 后面为增加后或减少后达到的数字。

5. 数词在句子中的作用

1）作主语

请看下例：

【例】The second is better.

第二个更好一些。

【例】Three nines are twenty-seven.

三九二十七。

句中，nines 是数词作主语。

【例】Three-fourths of the earth's surface is covered with water.

地球表面的四分之三被水覆盖。

【例】One is here, the other two are on that desk.

一个在这里，另外两个在那张桌子上。

2）作表语

请看下例：

【例】Five times five is twenty-five.

五乘以五等于二十五。

句中，twenty-five 是数词作表语。

【例】He was the first to get to the Great Wall.

他第一个到达长城。

【例】That man was the first to use a computer to solve this kind of problems.

那个人第一个用计算机来解决这个问题。

3）作宾语

请看下例：

【例】How many printers would you buy? Two.

你要买几台打印机？两台。

【例】That city has a population of five million.

这个城市有五百万人口。

【例】They use one-third of the money for new machines.

他们用三分之一的资金买新机器。

【例】They will discuss the design at two.

他们将在两点钟讨论这个设计。

4）作定语

请看下例：

【例】There are ten people in the room.

房间里有十个人。

【例】Only two printers were bought that day.

那天只买了两台打印机。

【例】Please press the second button on the right.

请按右边的第二个按钮。

Exercises

一、根据课文内容，判断以下叙述的正误。

(1) Software engineering is the process of manufacturing software systems.

(2) All coding today is done in one of the higher-level languages.

(3) A programmer typically writes the code using a text editor.

(4) Testing is the process of examining a software product to find errors.

(5) A test case type consists of the input values for the test.

(6) Large software systems are static.

(7) The maintenance phase may cost more than all of the others combined.

(8) The Y2K（千年虫问题）problem was not a maintenance problem.

(9) Software design is a process of problem-solving and planning for a software solution.

(10) A software design must be platform-independent.

二、根据课文内容填空。

(1) A software system consists of ____________________ and ____________________ needed to manufacture, use, and maintain the code.

(2) The purpose of the requirements phase is ________________ and ________________.

(3) The first phase of the software testing process is ________________ developed by a single programmer. The second phase is ________________ where units are combined and tested as a group. System testing is done ________________, usually with test cases developed from the system requirements. Acceptance testing of the system is done by ____________________.

(4) Higher-level languages can be categorized as ________________, ________________, and ____________________.

(5) The process of rerunning test cases to ensure that no error has been introduced is called ____________________.

(6) Maintenance consists of three activities: ________________, ________________, and ____________________.

(7) Systematic requirements analysis is also known as __________________________.

(8) Requirements must be ________________, ________________, ________________, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

(9) SRA stands for ______________________________.

(10) There are many aspects to consider in the design of a piece of software. Some of these aspects are __________________, __________________, __________________, __________________, __________________, __________________, __________________, __________________ and __________________.

三、用英语表示出下列数词。

(1) 1,234

(2) 567,345

(3) 39,632,851

(4) 187,238,465

(5) 58%

(6) 56.02

(7) 第 286

(8) 第 1001

(9) $3\frac{1}{2}$

(10) 5/6

四、阅读下列英文短句，选择合适的答案。

(1) Overflow in random access devices refers to ____________________.

A. linking

B. pumping

C. exception record

D. searching

(2) Block size on magnetic tape is generally limited by the ____________________.

A. length of the tape

B. number of records per block

C. limitations of internal storage capacity

D. density of the tape

(3) Magnetic tape read errors are most often corrected by ____________________.

A. cleaning the tape

B. rewinding the tapes and rereading record

C. backspacing tape and rereading record

D. change tape drives

(4) Which of the following is not a significant factor in determining the average access time of a mass storage device?

A. transfer rate

B. number of heads

C. average instruction execution time

D. rotational speed

(5) Simultaneous reception and transmission over a communications channel indicates which of the following?

A. time-sharing

B. simplex operation

C. half duplex operation

D. full duplex operation

(6) Which of the following is a technique by which each of the terminals sharing a communications line is periodically interrogated to determine if it requires servicing?

A. polling

B. priority scheduling

C. exclusive branching

D. synchronous operation

(7) To display the flow of program execution, a programmer would use ____________.

A. test checking

B. a trace routine

C. a program listing

D. The run manual

(8) Full memory dumps are most helpful when debugging ____________programs?

A. COBOL

B. Assembly

C. ALGOL

D. FORTRAN

(9) Which of the following would be used in a recursive program?

A. A push-down-pop-up stack

B. Reentrant coding

C. A subroutine

D. None of the above

(10) A subroutine must be passed parameters by ____________.

A. the calling program

B. the called program

C. the library routine

D. relocation

五、听短文，在画线处填写所听到的单词或词组。

The MP3 movement is one of the most amazing phenomena that the music industry has ever seen. Unlike other movements--for example, the ___1___ of the cassette tape or the CD — the MP3 movement started not with the industry itself but with a huge audience of 2) ___2___ on the

Internet. The MP3 format for ___3___ has had, and will continue to have, a huge impact on how people collect, listen to and ___4___ music.

Not everyone is happy with the rise in ___5___ of the MP3 format. Some audio enthusiasts say that most MP3 files can't ___6___ a CD or vinyl album version of the same song. Others go so far as to claim that the way ___7___ mix music is changing because of MP3s, and not necessarily in a good way.

If you have ever wondered how MP3 files work, or if you ___8___ MP3 files and wondered how to use them yourself, then this article is for you! In this article, you will learn about the MP3 file format and how you can start ___9___, listening to and ___10___ MP3 files onto CDs!

六、计算机软件水平考试真题自测（系统分析师级）：选择填空。

To compete in today's fast-paced competitive environment, organizations are increasingly allowing contractors, partners, visitors and guests to access their internal enterprise networks. These users may connect to the network through wired ports in conference rooms or offices, or via wireless access points. In allowing this open access for third parties, LANs become ___A___. Third parties can introduce risk in a variety of ways from connecting with an infected laptop to unauthorized access of network resources to ___B___ activity. For many organizations, however, the operational complexity and costs to ensure safe third party network access have been prohibitive. Fifty-two percent of surveyed CIOs state that they currently use a moat and castle's security approach, and admit that defenses inside the perimeter are weak. Threats from internal users are also increasingly a cause for security concerns. Employees with malicious intent can launch ___C___ of service attacks or steal ___D___ information by snooping the network. As they access the corporate network, mobile and remote users inadvertently can infect the network with ___E___ and worms acquired from unprotected public networks. Hackers masquerading as internal users can take advantage of weak internal security to gain access to confidential information.

供选择的答案：

A.（1）damageable （2）susceptible （3）vulnerable （4）changeable

B.（1）venomous （2）malicious （3）felonious （4）villainous

C.（1）denial （2）virtuous （3）renounce （4）traverse

D.（1）reserved （2）confidential （3）complete （4）mysterious

E.（1）sickness （2）disease （3）viruses （4）germs

Skill Training

如何写软件产品的介绍

简单软件产品介绍或软件产品说明书，通常包括以下几部分。

1. 软件产品概述

软件产品概述包括：软件名、开发商、版本信息、软件的背景知识、业务特色、技术特色，

主要功能等。

2. 运行环境

（1）对硬件的要求：显卡、内存及 CPU 等的说明。
（2）对网络的要求：是否需要在有网络的情况才可使用，等等。
（3）对软件环境的要求：是运行在客户端操作系统上（如 windows XP/7），还是运行在服务器操作系统上（Windows Server/Unix/Linux）。

3. 如何安装软件

说明如何安装该软件，以及在安装过程中需要注意的问题等。

4. 软件功能的说明

软件功能说明要尽量详细，包括：
总体功能的概述；
子系统 1 的功能说明｛模块 1 功能说明，模块 2 功能说明……模块 m 的功能说明｝；
子系统 2 的功能说明｛模块 1 功能说明，模块 2 功能说明……模块 m 的功能说明｝；
⋮
子系统 n 的功能说明｛模块 1 功能说明，模块 2 功能说明……模块 m 的功能说明｝。

5. 技术支持以及联系方式

技术支持以及联系方式通常包括：公司地址，网址，E-mail，电话等。

6. 范例

Skype

Description

Skype is the always-on app that makes staying in touch with all your contacts easier than ever.

Stay informed about what's happening with your friends through video and voice calls and instant messaging, all from one app. So, no matter if you're on a PC, Windows Phone, Mac, tablet, iPhone or Android device, Skype keeps you connected and always available.

Features

- Once you sign in, the app is always on without draining your device's battery, so you can take calls any time.
- Exchange instant messages with all your Skype, Messenger and Outlook. com contacts.
- Call any phone quickly and easily from Skype.
- All Skype-to-Skype video and voice calls are free.
- Your messages are delivered straight to you no matter what you're doing.
- Snap Skype left or right and do more while you IM, video call or keep an eye on your home screen.

- Share photos, videos and files of any size over Skype.

Supported processors
x86, x64, ARM

Languages

English (United States), Chinese (China), Arabic, German, Spanish, Japanese, French, Russian, Portuguese, Bulgarian, Croatian, Danish, Estonian, Finnish, Greek, Chinese (Hong Kong SAR), Hebrew, Italian, Korean, Latvian, Lithuanian, Dutch, Polish, Portuguese (Brazil), Romanian, Slovak, Swedish, Chinese (Taiwan), Thai, Turkish, Ukrainian, English (United Kingdom), Serbian (Latin), Slovenian, Hungarian, Catalan, Czech, Indonesian, Norwegian (Bokm l), Vietnamese

Contact

To contact Skype in relation to the "Skype" branded software or products, please submit a support request to customer support team. To contact Skype in relation to the "Qik" branded software or products, please submit a support request to http://support.qik.com

For members of Skype Manager in the United States only:

Skype Inc.

3210 Porter Drive

Palo Alto, California 94304, USA

Reading Material

Unified Modeling Language[1]

In the field of software engineering, the Unified/Universal Modeling Language (UML) is a standardized visual specification language for object modeling. UML is a general-purpose modeling language that includes a graphical notation used to create an abstract model of a system, referred to as a UML model.

1. General description

UML is officially defined at the Object Management Group (OMG)[2] by the UML metamodel, a Meta-Object Facility (MOF)[3] metamodel. Like other MOF-based specifications, UML has allowed software developers to concentrate more on design and architecture.

[1] Unified Modeling Language (UML) 统一建模语言
[2] Object Management Group (OMG) 对象管理组
[3] Meta-Object Facility (MOF) 元对象工具

UML models may be automatically transformed to other representations (e.g., Java) by means of QVT[1]-like transformation languages, supported by the OMG.

UML is extensible, offering the following mechanisms for customization[2]: profiles and stereotype[3]. The semantics of extension by profiles have been improved with the UML 2.0 major revision[4].

2. History

After Rational Software Corporation hired James Rumbaugh from General Electric in 1994, the company became the source for the two most popular object-oriented modeling approaches of the day: Rumbaugh's OMT that was better for object-oriented analysis (OOA), and Grady Booch's Booch method that was better for object-oriented design (OOD). Together Rumbaugh and Booch attempted to reconcile[5] their two approaches and started work on a Unified Method.

They were soon assisted in their efforts by Ivar Jacobson, the creator of the OOSE method. Jacobson joined Rational in 1995, after his company, Objectory, was acquired by Rational. The three methodologists[6] were collectively referred to as the Three Amigos, since they were well known to argue frequently with each other regarding methodological preferences.

In 1996 Rational concluded that the abundance[7] of modeling languages was slowing the adoption of object technology, so repositioning[8] the work on a Unified Method, they tasked the Three Amigos with the development of a non-proprietary Unified Modeling Language. Representatives of competing Object Technology companies were consulted during OOPSLA'96, and were won over by Rumbaugh's cappella rendition of his version of Joni Mitchell's "Clouds", indicating the victory of his OMT notation of using boxes for representing classes over Grady Booch's Booch method's notation that used cloud symbols[9].

Under the technical leadership of the Three Amigos, an international consortium called the UML Partners was organized in 1996 to complete the Unified Modeling Language (UML) specification, and propose it as a response to the OMG RFP. The UML Partners' UML 1.0 specification draft[10] was proposed to the OMG in January 1997. During the same month the UML Partners formed a Semantics Task Force, chaired by Cris Kobryn and administered by Ed Eykholt, to finalize[11] the semantics of the specification and integrate it with other standardization efforts. The

[1] QVT (Query View Transformation) 查询/视图/转换
[2] customization [ˌkʌstəmaiˈzeiʃən] *n.* 用户化，专用化，定制
[3] stereotype [ˈstiəriəutaip] *vt.* 使定型，使固定
[4] revision [riˈviʒən] *n.* 修订，修改，修正，修订本
[5] reconcile [ˈrekənsail] *vt.* 使和解，使和谐
[6] methodologist [ˌmeθəˈdɔlədʒist] *n.* 方法论学者，方法学家
[7] abundance [əˈbʌndəns] *n.* 丰富，充裕
[8] reposition [ˌriːpəˈziʃən] *n.* 重新配置
v. 保存，存放
[9] symbol [ˈsimbəl] *n.* 符号，记号
[10] draft [drɑːft] *n.* 草稿，草案，草图
[11] finalize [ˈfainəlaiz] *v.* 把（计划、稿件等）最后定下来，定案

result of this work, UML 1.1, was submitted[1] to the OMG in August 1997 and adopted by the OMG in November 1997.

As a modeling notation, the influence of the OMT notation dominates[2] (e.g., using rectangles for classes and objects). Though the Booch "cloud" notation was dropped, the Booch capability to specify lower-level design detail was embraced. The use case notation from Objectory and the component notation from Booch were integrated with the rest of the notation, but the semantic integration was relatively weak in UML 1.1, and was not really fixed[3] until the UML 2.0 major revision.

Concepts from many other OO methods were also loosely integrated with UML with the intent that UML would support all OO methods. For example, CRC Cards (circa 1989 from Kent Beck and Ward Cunningham) and OORam were retained[4]. Many others contributed too with their approaches flavoring the many models of the day including: Tony Wasserman and Peter Pircher with the "Object-Oriented Structured Design (OOSD)[5]" notation (not a method), Ray Buhr's "Systems Design with Ada", Archie Bowen's use case and timing analysis, Paul Ward's data analysis and David Harel's "Statecharts", as the group tried to ensure broad coverage in the real-time systems domain. As a result, UML is useful in a variety of engineering problems, from single process, single user applications to concurrent, distributed systems[6], making UML rich but large.

The Unified Modeling Language is an international standard:

ISO/IEC 19501: 2005 Information technology—Open Distributed Processing—Unified Modeling Language (UML) Version 1.4.2.

UML has matured significantly since UML 1.1. Several minor revisions (UML 1.3, 1.4 and 1.5) fixed shortcomings[7] and bugs with the first version of UML, followed by the UML 2.0 major revision that was adopted by the OMG in 2003. There are four parts to the UML 2.x specification: the Superstructure that defines the notation and semantics for diagrams and their model elements; the Infrastructure that defines the core metamodel on which the Superstructure is based; the Object Constraint Language (OCL)[8] for defining rules for model elements; and the UML Diagram Interchange that defines how UML 2 diagram layouts are exchanged. The current versions of these standards follow: UML Superstructure version 2.1.2, UML Infrastructure version 2.1.2, OCL version 2.0, and UML Diagram Interchange version 1.0.

Although many UML tools support some of the new features of UML 2.x, the OMG provides no test suite to objectively test compliance with its specifications.

---

[1] submit [səbˈmit] *vt.* 提交，递交
[2] dominate [ˈdɔmineit] *v.* 支配，占优势
[3] fixed [fikst] *adj.* 固定的，确定的
[4] retain [riˈtein] *vt.* 保持，保留
[5] Object-Oriented Structured Design (OOSD) 面向对象的结构化设计
[6] distributed system 分布式系统
[7] shortcoming [ˈʃɔːtkʌmiŋ] *n.* 缺点，短处
[8] Object Constraint Language (OCL) 对象约束语言

3. Methods

UML is not a method by itself; however, it was designed to be compatible with the leading object-oriented software development methods of its time (for example OMT, Booch, Objectory). Since UML has evolved, some of these methods have been recast[1] to take advantage of the new notation (for example OMT), and new methods have been created based on UML. The best known is Rational Unified Process (RUP). There are many other UML-based methods like Abstraction Method, Dynamic Systems Development Method, and others, designed to provide more specific solutions, or achieve different objectives.

4. Modeling

It is very important to distinguish between the UML model and the set of diagrams of a system. A diagram is a partial graphical representation of a system's model. The model also contains a "semantic backplane" — documentation such as written use cases[2] that drive the model elements and diagrams.

UML diagrams represent three different views of a system model:

Functional requirements view: Emphasizes the functional requirements of the system from the user's point of view. Includes use case diagrams.

Static structural view: Emphasizes the static structure of the system using objects, attributes, operations and relationships. Includes class diagrams[3] and composite[4] structure diagrams.

Dynamic behavior view: Emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. Includes sequence diagrams[5], activity diagrams[6] and state machine diagrams.

UML models can be exchanged among UML tools by using the XMI[7] interchange format.

5. Diagrams

UML 2. 0 has 13 types of diagrams that can be categorized hierarchically as shown in the following class diagram.

(1) Structure diagrams[8] emphasize what things must be in the system being modeled:

- Class diagram;
- Component diagram[9];
- Composite structure diagram (added in UML 2. x);

---

[1] recast [ˈriːˈkɑːst] *v.* 重铸，彻底改动，重做
[2] case [keis] *n.* 用例，案例
[3] class diagram 类图
[4] composite [ˈkɔmpəzit] *adj.* 合成的，复合的
[5] sequence diagram 序列图
[6] activity diagram 活动图
[7] XMI (XML Metadata Interchange) XML 元数据交换
[8] structure diagram 结构图
[9] component diagram 构件图

- Deployment diagram[1];
- Object diagram;
- Package diagram[2].

(2) Behavior diagrams emphasize what must happen in the system being modeled:

- Activity diagram;
- State machine diagram;
- Use case diagram.

(3) Interaction diagrams, a subset of behavior diagrams, emphasize the flow of control and data among the things in the system being modeled:

- Communication diagram;
- Interaction overview diagram (added in UML 2. x);
- Sequence diagram;
- Timing diagram (added in UML 2. x).

The Protocol State Machine is a sub-variant of the State Machine. It may be used to model network communication protocols.

UML does not restrict UML element types to a certain diagram type. In general, every UML element may appear on almost all types of diagrams. This flexibility has been partially restricted in UML 2.0.

In keeping with the tradition of engineering drawings, a comment or note explaining usage, constraint, or intent is always allowed in a UML diagram.

6. Criticisms

Although UML is a widely recognized and used modeling standard, it is frequently criticized for the following deficiencies:

- Language bloat[3]. UML is often criticized as being gratuitously[4] large and complex. It contains many diagrams and constructs that are redundant or infrequently used. This criticism is more frequently directed at UML 2.0 than UML 1.0, since newer revisions include more design-by-committee compromises[5].
- Problems in learning and adopting. The problems cited above can make learning and adopting UML problematic, especially when required of engineers lacking the prerequisite[6] skills.
- Only the code is in sync with the code. Another perspective holds that it is working systems that are important, not beautiful models. As Jack Reeves succinctly[7] put it, "The code is the design." Pursuing this notion leads to the need for better ways of writing software; UML

[1] deployment diagram 部署图
[2] package diagram 包图
[3] bloat [bləut] *v.* 膨胀
[4] gratuitously [grəˈtjuːitəsli] *adv.* 无理由地，无代价地
[5] compromise [ˈkɔmprəmaiz] *n.* 妥协，折中
[6] prerequisite [priːˈrekwizit] *adj.* 必须具备的，先决条件的
[7] succinctly [səkˈsiŋktli] *adv.* 简洁地，简便地

has value in approaches that compile the models to generate source or executable code. This however, may still not be sufficient since it is not clear that UML 2.0's Action Semantics exhibit[1] Turing completeness[2]. "All models are wrong, but some models are useful."

- Cumulative[3] Impedance/Impedance Mismatching. As with any notational system, UML is able to represent some systems more concisely or efficiently than others. Thus a developer gravitates toward solutions that reside at the intersection of the capabilities of UML and the implementation language. This problem is particularly pronounced if the implementation language does not adhere to orthodox[4] object-oriented doctrine[5], as the intersection set[6] between UML and implementation language may be that much smaller.
- Aesthetically Inconsistent. This argument states that the adhoc mixing of abstract notation (2-D ovals, boxes, etc) make UML appear jarring and that more effort could have been made to construct uniform and aesthetically pleasing representations.
- Tries to be all things to all programmers. UML is a general purpose modeling language that tries to achieve compatibility with every possible implementation language. In the context of a specific project, the most applicable features of UML must be delimited[7] for use by the design team to accomplish the specific goal. Additionally, the means of restricting the scope of UML to a particular domain is through a formalism[8] that is not completely formed, and is itself the subject of criticism.
- Dysfunctional[9] interchange format. While the XMI (XML Metadata Interchange) standard is designed to facilitate the interchange of UML models, it has been largely ineffective[10] in the practical interchange of UML 2.x models. Defining a UML 2.x model in one tool and then importing it into another tool typically leads to loss of information. This interoperability ineffectiveness is attributable to two reasons: First, XMI 2.x is large and complex in its own right, since it purports to address a technical problem more ambitious than exchanging UML 2.x models. In particular, it attempts to provide a mechanism for facilitating the exchange of any arbitrary modeling language defined by the OMG's Meta-Object Facility (MOF). Secondly, the UML 2.x Diagram Interchange specification lacks sufficient detail to facilitate reliable interchange of UML 2.x notations between modeling tools. Since UML is a visual modeling language[11], this shortcoming is substantial for modelers who don't want to redraw their diagrams.

---

[1] exhibit [ig'zibit] *v.* 展示
[2] completeness [kəm'pli:tnis] *n.* 完备性
[3] cumulative ['kju:mjulətiv] *adj.* 累积的
[4] orthodox ['ɔ:θədɔks] *adj.* 正统的，传统的，习惯的
[5] doctrine ['dɔktrin] *n.* 教条，学说
[6] intersection set 交集
[7] delimit [di'limit] *vt.* 限定，划定……界限
[8] formalism ['fɔ:məlizəm] *n.* 拘泥形式，形式主义
[9] dysfunctional [dis'fʌŋkʃənl] *adj.* 功能失常的
[10] ineffective [ini'fektiv] *adj.* 无效的，无能力的
[11] visual modeling language 可视化建模语言

参 考 译 文

软 件 工 程

1. 软件工程基础

软件工程是生产软件系统的过程。软件系统包括可执行的计算机代码和制造、使用和维护这些代码所需要的支持文档。例如，文字处理系统包括可执行程序（字处理器）、用户手册和诸如需求、设计及产生可执行程序和手册所需的文档。

随着更大、更复杂的及在生活中尤为重要的软件系统的迅速增长，软件工程越加重要。计算机硬件成本的快速降低意味着典型系统中软件的成本比所用硬件的成本更高。大的软件系统可能比过去更复杂。这就极大地要求对软件工程过程需要实行严格管理和控制。

软件工程过程自身通常被分为若干阶段。这些阶段的定义、它们的顺序及这些阶段之间的交互指定了软件生命周期模型。著名的生命周期模型是瀑布模型，它包括了需求定义、设计、编码、测试和维护各阶段。每个阶段的输出作为下一阶段的输入。

需求阶段的目的是定义系统要做什么以及它运行的约束条件。这些信息记录在需求文档中。一个典型的需求文档可能包括产品概述、开发规范、产品的运行和维护环境、系统的高级概念模型、用户界面规范、功能需求规范、非功能需求规范、系统外部接口规范、如何处理错误的规范以及系统可能改变和增强的清单。每个需求，通常都标有基准，必须是可测试的。

在设计阶段，制定系统如何实现这些需求的计划。该计划用设计方法和符号表示。现在已经开发出了用于软件设计的许多方法和符号。每个方法都注重系统的某些方面而忽略或最小化其他方面。这类似于用建筑图、管路图、电路图等来看一个建筑。

软件生命周期的编码阶段与开发实现设计的代码相关。这个代码用一种正式的称为编程语言的语言写成。编程语言已经经历了直接由计算机解释的 0 和 1 序列，由符号机器码、汇编语言到最终越来越容易被人们所理解的高级语言阶段。

如今大部分编码都是用一种高级语言编写的。当代码用高级语言写成后，它就被编译器翻译成汇编代码，并最终译为机器码。现在人们已经开发出了许多高级语言，它们可以分为函数语言、说明语言和强制语言。

根据模块化原则，大系统中的代码被分成模块，并且把这些模块分配给各个程序员。程序员通常用文本编辑器写代码。有时也使用语法导向的编辑器，它“了解”特定的编程语言并提供编程模版，而且能检查语法错误。程序员也可以使用其他多种工具，包括帮助寻找代码中错误的调试器、表明模块各部分花费的最长执行时间的探查器以及使代码运行得更快的优化器。

测试是对软件产品进行检查以便发现错误的过程。不仅代码需要测试，而且所有生命周期产品和支持该软件的文档（如用户手册）都需要测试。

软件测试过程经常可以分为几个阶段。第一阶段是单元测试单个程序员所开发的软件。第二阶段是把单元组合然后按组测试的集成测试。系统测试针对整个系统，通常使用根据系

统需求开发的测试用例。系统的验收测试由预期用户完成。

测试的基本单位是测试用例。测试用例由以下几部分组成：测试用例类型——测试用例应该运行的系统方面；测试条件——由测试的输入值组成；用在测试中系统的环境状态和在给定输入和环境因素下的系统预期行为。

因为修正错误或增加功能而改变软件时，通常会招制严重的错误。要确保不发生这样的事情，每次改变软件后就必须重新运行所有的测试用例。这个重新运行测试用例以保证不会引入错误的过程称为回归测试。

使用预排和检查提高软件开发过程的质量。因此，通过这一步骤建立的软件产品的质量也得以改进。质量系统是各种技术的集合，应用这些技术会不断改进产品在软件开发过程中的质量。质量系统的元素包括评审、检查和过程审核。

大的软件系统不是静态的，而是在开发期间和使用之后频繁改变的。维护是使用之后的软件生命周期的阶段。维护阶段的成本也许比其他所有阶段的总和还要多，因此软件组织也十分重视它。例如，千年虫问题就是一个维护问题。

维护包括三种行为：适应、修正和增加。增加是给系统增加新功能的过程，通常根据系统用户的请求进行。这个行为需要有自己完整的生命周期。也就是说，增加新功能要有需求、设计、实现和测试。研究表明有大约一半的维护是增加维护。

适应性维护是改变系统以便使其适应新环境的过程，例如把一个系统从 Windows 操作系统移植到 Linux 操作系统。已经发现适应性维护占维护总量的 1/4。更正性维护是修改发布后系统中错误的过程。更正性维护占维护总量的 20%。

因为操作系统随着时间频繁改变。一个重要的活动是软件配置管理。这包括跟踪生命周期目标版本、控制它们的变化以及监控它们的关系。配置管理活动包括版本控制——跟踪生命周期目标版本；变化控制——处理系统改变请求的一系列过程；建造控制——跟踪与工作产品一起运行的版本以便形成软件产品的给定版本。

2. 需求分析

在软件工程中，需求分析包括以下任务：探究需要满足新的或已经改变的设备的需求或条件、估计不同股东（如收益人或用户）间可能的需求冲突。需求分析对于软件项目的成功开发十分重要。

系统需求分析也称为需求工程。有时也使用更宽泛的名称，如需求收集、需求捕捉或需求规范。

需求必须是可行动的、可测量的、可测试的，与指定的商业需求或机会相关，并对系统设计进行足够详细的定义。

3. 软件设计

软件设计是解决问题和规划软件解决方案的过程。确定了软件的规范和目标后，软件开发者将自己设计或雇用设计师来制订解决计划。它包括低级部件和算法实施问题以及结构图。

软件开发过程的软件需求分析（SRA）步骤产生软件工程中所使用的规范。如果该软件是“半自动的”或“以用户为中心的”，软件设计可以包括用户体验设计，它产生一个情节

串联图板以帮助确定这些规范。如果该软件是完全自动化的（意味着没有用户或用户接口），那么软件设计就可以如画流程图或文本来描述计划好的一系列事件那样简单。也有半自动的方法，如统一建模语言和基本建模概念。在其中任意情况下，计划的某些文档通常是设计的成果。

软件设计可以是独立平台的，也可以是专用平台的，这取决于设计所用的技术。

在设计一个软件时有许多方面需要考虑。每个方面都很重要，它应该反映该软件要实现的目标。这些方面有：

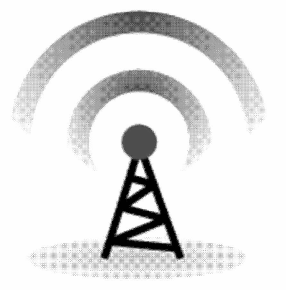
- 可扩展性——无须对结构体系进行大的改变就可以给软件增加新的性能。
- 健壮性——在重压或不可预知或无效输入的情况下，软件仍然可以运行。例如，软件可以被设计为具有弹性以适应低内存条件。
- 可靠性——在特定时期内，在给定条件下，软件可以执行必要的功能。
- 容错性——软件可以抵抗部件失效并从中恢复。
- 安全性——软件可以经受敌对行为和影响。
- 可维护性——软件可以在指定时期内恢复到指定的状态。例如，抗病毒软件可以定时接收病毒定义更新以便维护软件的效用。
- 兼容性——软件能够与其他设计用来协同工作的产品一起运行。例如，一个软件可以向后兼容自己的老版本。
- 模块性——最终的软件由定义好的独立部件组成。这就产生了良好的可维护性。然后，在集成到期望的软件产品之前，可以执行这些部件或进行独立测试。这就允许软件在开发中各部分工作分别进行。
- 复用性——所设计的模块部件应该具有正好所期望的功能要素。这种单用途的部件可以在其他类似设计中再次使用。

Lesson 12

Text

Wireless Network

Wireless network refers to any type of computer network that uses wireless (usually, but not always) radio waves for network connections.


Wireless icon

It is a method by which homes, telecommunications networks and enterprise (business) installations avoid the costly process of introducing cables into a building, or as a connection between various equipment locations. Wireless telecommunications networks are generally implemented and administered using radio communication. This implementation takes place at the physical layer of the OSI model network structure.

1. Types of wireless networks

1.1 Wireless PAN

Wireless personal area networks (WPANs) interconnect devices within a relatively small area, which is generally within a person's reach. For example, both Bluetooth radio and invisible infrared light provides a WPAN for interconnecting a headset to a laptop. ZigBee also supports WPAN applications. Wi-Fi PANs are becoming commonplace (2010) as equipment designers start to integrate Wi-Fi into a variety of consumer electronic devices. Intel "My Wi-Fi" and Windows 7 "virtual Wi-Fi" capabilities have made Wi-Fi PANs simpler and easier to set up and configure.

1.2 Wireless LAN

A wireless local area network (WLAN) links two or more devices over a short distance using a wireless distribution method, usually providing a connection through an access point for Internet access. The use of spread spectrum or OFDM (Orthogonal frequency-division multiplexing) technologies may allow users to move around within a local coverage area, and still remain connected to the network.

Products using the IEEE 802.11 WLAN standards are marketed under the Wi-Fi brand name. Fixed wireless technology implements point-to-point links between computers or networks at two

distant locations, often using dedicated microwave or modulated laser light beams over line of sight paths. It is often used in cities to connect networks in two or more buildings without installing a wired link.

1.3 Wireless mesh network

A wireless mesh network is a wireless network made up of radio nodes organized in a mesh topology. Each node forwards messages on behalf of the other nodes. Mesh networks can "self heal", automatically re-routing around a node that has lost power.

1.4 Wireless MAN

Wireless metropolitan area networks are a type of wireless network that connects several wireless LANs.

WiMAX is a type of Wireless MAN and is described by the IEEE 802.16 standard.

1.5 Wireless WAN

Wireless wide area networks are wireless networks that typically cover large areas, such as between neighboring towns and cities, or city and suburb. These networks can be used to connect branch offices of business or as a public internet access system. The wireless connections between access points are usually point to point microwave links using parabolic dishes on the 2.4 GHz band, rather than omnidirectional antennas used with smaller networks. A typical system contains base station gateways, access points and wireless bridging relays. Other configurations are mesh systems where each access point acts as a relay also. When combined with renewable energy systems such as photovoltaic solar panels or wind systems they can be stand-alone systems.

1.6 Cellular network

A cellular network or mobile network is a radio network distributed over land areas called cells, each served by at least one fixed-location transceiver, known as a cell site or base station. In a cellular network, each cell characteristically uses a different set of radio frequencies from all their immediate neighbouring cells to avoid any interference.

When joined together these cells provide radio coverage over a wide geographic area. This enables a large number of portable transceivers (e.g., mobile phones, pagers, etc.) to communicate with each other and with fixed transceivers and telephones anywhere in the network, via base stations, even if some of the transceivers are moving through more than one cell during transmission.

Although originally intended for cell phones, with the development of smartphones, cellular telephone networks routinely carry data in addition to telephone conversations:

- Global System for Mobile Communications (GSM): The GSM network is divided into three major systems: the switching system, the base station system, and the operation and support system. The cell phone connects to the base system station which then connects to the

operation and support station; it then connects to the switching station where the call is transferred to where it needs to go. GSM is the most common standard and is used for a majority of cell phones.

- Personal Communications Service (PCS): PCS is a radio band that can be used by mobile phones in North America and South Asia. Sprint happened to be the first service to set up a PCS.
- D-AMPS: (Digital AMPS) Digital Advanced Mobile Phone Service, an upgraded version of AMPS (Advanced Mobil Phone Service), is being phased out due to advancement in technology. The newer GSM networks are replacing the older system.

2. Uses

Some examples of usage include cellular phones which are part of everyday wireless networks, allowing easy personal communications. Another example, Intercontinental network systems, used radio satellites to communicate across the world. Emergency services such as the police utilize wireless networks to communicate effectively as well. Individuals and businesses use wireless networks to send and share data rapidly, whether it be in a small office building or across the world.

3. Properties

3.1 General

In a general sense, wireless networks offer a vast variety of uses by both business and home users.

"Now, the industry accepts a handful of different wireless technologies. Each wireless technology is defined by a standard that describes unique functions at both the Physical and the Data Link layers of the OSI Model. These standards differ in their specified signaling methods, geographic ranges, and frequency usages, among other things. Such differences can make certain technologies better suited to home networks and others better suited to network larger organizations."

3.2 Performance

Each standard varies in geographical range, thus making one standard more ideal than the next depending on what it is one is trying to accomplish with a wireless network. The performance of wireless networks satisfies a variety of applications such as voice and video. The use of this technology also gives room for expansions, such as from 2G to 3G and, most recently, 4G technology, which stands for fourth generation of cell phone mobile communications standards. As wireless networking has become commonplace, sophistication increases through configuration of network hardware and software, and greater capacity to send and receive larger amounts of data faster is achieved.

3.3 Space

Space is another characteristic of wireless networking. Wireless networks offer many advantages when it comes to difficult-to-wire areas trying to communicate such as across a street or river, a warehouse on the other side of the premise or buildings that are physically separated but operate as one. Wireless networks allow for users to designate a certain space which the network will be able to communicate with other devices through that network. Space is also created in homes as a result of eliminating clutters of wiring. This technology allows for an alternative to installing physical network mediums such as TPs (Twisted pairs), COAXes (Coaxial Cable), or fiber optics, which can also be expensive.

3.4 Home

For homeowners, wireless technology is an effective option compared to Ethernet for sharing printers, scanners, and high speed internet connections. WLANs help save the cost of installation of cable mediums, save time from physical installation, and also create mobility for devices connected to the network. Wireless networks are simple and require as few as one single wireless access point connected directly to the Internet via a router.

3.5 Wireless Network Elements

The telecommunications network at the physical layer also consists of many interconnected wireline Network Elements (NEs). These NEs can be stand-alone systems or products that are either supplied by a single manufacturer, or are assembled by the service provider or system integrator with parts from several different manufacturers.

Reliable wireless service depends on the network elements at the physical layer to be protected against all operational environments and applications.

What are especially important are the NEs that are located on the cell tower to the Base Station (BS) cabinet. The attachment hardware and the positioning of the antenna and associated closures/cables are required to have adequate strength, robustness, corrosion resistance, and rain/solar resistance for expected wind, storm, ice, and other weather conditions. Requirements for individual components, such as hardware, cables, connectors, and closures, shall take into consideration the structure to which they are attached.

3.6 Capacity

3.6.1 Link

The maximum data rate of any single wireless link can be described by the Shannon's theorem which is related to the bandwidth in hertz, and the noise on the channel.

3.6.2 Network

The total network bandwidth depends on how dispersive the medium is (more dispersive medium generally has better total bandwidth because it minimizes interference), how many

frequencies are available, how noisy those frequencies are, whether directional antenna are in use, whether nodes employ power control and so on.

New Words

installation	[ˌinstəˈleiʃən]	*n.* 安装，装置
costly	[ˈkɔstli]	*adj.* 昂贵的，贵重的
administer	[ədˈministə]	*v.* 管理，执行
connective	[kəˈnəktiv]	*adj.* 连接的
cable	[ˈkeibl]	*n.* 电缆
infrared	[ˈinfrəˈred]	*adj.* 红外线的
		n. 红外线
commonplace	[ˈkɔmənpleis]	*n.* 平凡的事，平常话
		adj. 平凡的
virtual	[ˈvəːtjuəl]	*adj.* 虚拟的
configure	[kənˈfigə]	*vi.* 配置，设定
microwave	[ˈmaikrəuweiv]	*n.* 微波
modulated	[ˈmɔdjuleitid]	*adj.* 已调制的，被调的
band	[bænd]	*n.* 波段
omnidirectional	[ˌɔmnidiˈrekʃənl]	*adj.* 全方向的
relay	[ˈriːlei]	*n.* 转发器
		vt. 使接替，转发
configuration	[kənˌfigjuˈreiʃən]	*n.* 构造，结构，配置，外形
renewable	[riˈnju(ː)əbl]	*adj.* 可更新的，可恢复的
photovoltaic	[ˌfəutəuvəlˈteiik]	*adj.* 光电的
immediate	[iˈmiːdjət]	*adj.* 紧接的，紧靠的
coverage	[ˈkʌvəridʒ]	*n.* 覆盖
pager	[ˈpeidʒə]	*n.* 寻呼机
smartphone	[ˈsmɑːtfəun]	*n.* 智能电话
conversation	[ˌkɔnvəˈseiʃən]	*n.* 会话，交谈
sophistication	[səˌfistiˈkeiʃən]	*n.* 先进性
intercontinental	[ˌintəˌkɔntiˈnentl]	*adj.* 大陆间的，洲际的
satellite	[ˈsætəlait]	*n.* 人造卫星
frequency	[ˈfriːkwənsi]	*n.* 频率，周率
designate	[ˈdezigneit]	*v.* 指定，指派
eliminate	[iˈlimineit]	*vt.* 排除，消除
clutter	[ˈklʌtə]	*n.* 混乱，杂乱
assemble	[əˈsembl]	*vt.* 组装

integrator	[ˈintigreitə]	*n*. 集成者，综合者
medium	[ˈmiːdjəm]	*n*. 介质，媒介
attachment	[əˈtætʃmənt]	*n*. 附件，附加装置
robustness	[rəˈbʌstnis]	*n*. 坚固性，健壮性；鲁棒性
connector	[kəˈnəktə]	*n*. 连接器
closure	[ˈkləuʒə]	*n*. 封闭器
channel	[ˈtʃænl]	*n*. 信道，频道
dispersive	[disˈpəːsiv]	*adj*. 分散的
interference	[ˌintəˈfiərəns]	*n*. 冲突，干涉

Phrases

wireless network	无线网络
radio wave	无线电波
telecommunications networks	通信网络
radio communication	无线电通讯
physical layer	物理层
integrate … into …	把……整合到……
a variety of	多种的
spread spectrum	扩频
coverage area	（电波）有效区
point-to-point link	点对点链接
laser light beam	激光束
line of sight	视线
wireless mesh network	无线网状网络
radio node	无线节点
mesh topology	网状网络
self heal	自愈
wireless metropolitan area networks	无线城域网
wireless wide area networks	无线广域网
access point	接入点，存取点
microwave link	微波接力线路
parabolic dish	抛物柱面反射器
base station	基点，基地，基站
wireless bridging relay	无线桥接中继
mesh system	网状系统
solar panel	太阳电池板
wind system	风力发电系统

stand-alone system	独立系统
cellular network	蜂窝网络
mobile network	移动网络
cell site	蜂窝基站
portable transceiver	便携式收发器
join together	连接，结合
be used for…	用来做……
a majority of	大部分
phase out	使逐步淘汰，逐渐停止
cellular phone	便携式电话
a handful of	一把，少数，少量
give room for…	有……余地
when it comes to	当谈论到
difficult-to-wire area	布线困难的区域，难以布线的区域
as a result of	作为结果
fiber optic	光纤
corrosion resistance	耐（腐）蚀性，耐蚀力，抗腐（蚀）性
take into consideration	考虑
Shannon's theorem	香农定理
directional antenna	有向天线

Abbreviations

OSI（Open System Interconnect ）	开放式系统互联
WPAN （Wireless personal area network）	无线个人区域网
WLAN（wireless local area network）	无线局域网
OFDM（Orthogonal frequency-division multiplexing）	正交频分复用技术
GSM（Global System for Mobile Communications）	全球通
PCS（Personal Communications Service）	个人通信服务
AMPS（Advanced Mobile Phone Service）	高级移动电话服务
D-AMPS（Digital AMPS）	数字高级移动电话服务
TP（Twisted pair）	双绞线
COAX （Coaxial Cable）	同轴电缆
NE（Network Element）	网络元件

Notes

[1] It is a method by which homes, telecommunications networks and enterprise (business) installations avoid the costly process of introducing cables into a building, or as a connection

between various equipment locations.

本句中，it 指上一段的 wireless network，by which 引导了一个定语从句，修饰和限定 a method。costly 是一个形容词，意思是“昂贵的”，introduce sth. into…的意思是“把……引入……”。

［2］A cellular network or mobile network is a radio network distributed over land areas called cells, each served by at least one fixed-location transceiver, known as a cell site or base station.

本句中，distributed over land areas 是一个过去分词短语，作定语，修饰和限定 a radio network，called cells 也是一个过去分词短语，作定语，修饰和限定 land areas。each served by at least one fixed-location transceiver, known as a cell site or base station 对 cells 进行解释说明。known as a cell site or base station 是对 each 进一步补充说明。

［3］The cell phone connects to the base system station which then connects to the operation and support station.

本句中，which then connects to the operation and support station 是一个定语从句，修饰和限定 the base system station。

［4］Each standard varies in geographical range, thus making one standard more ideal than the next depending on what it is one is trying to accomplish with a wireless network.

本句中，thus making one standard more ideal than the next depending on what it is one is trying to accomplish with a wireless network 是现在分词短语作结果状语，其中 depending on what it is one is trying to accomplish with a wireless network 作条件状语。

［5］What are especially important are the NEs that are located on the cell tower to the Base Station（BS）cabinet.

本句中，what are especially important 是一个主语从句，意思是“尤其重要的是”。that are located on the cell tower to the Base Station（BS）cabinet 是一个定语从句，修饰和限定 the NEs。

［6］The maximum data rate of any single wireless link can be described by the Shannon's theorem which is related to the bandwidth in hertz, and the noise on the channel.

本句中，which is related to the bandwidth in hertz, and the noise on the channel 是一个定语从句，修饰和限定 the Shannon's theorem。

Grammar

同位语和插入语

1. 同位语

1）同位语的定义

同位语用来对一个词或词组的内容加以说明。它通常位于与之同位的词或词组之后。请看下例：

【例】Our teacher, Mr. Smith, left for Beijing yesterday.

我们的老师，史密斯先生，昨天去北京了。

本句中，Mr. Smith 作 our teacher 的同位语。

【例】Word comes that he has developed a kind of new software.

有消息说他已经研制了一种新软件。

本句中，that he has developed a kind of new software 是一个从句，作 word 的同位语，说明 word 的内容。这种从句叫作同位语从句。

【例】What would you like us three to do for you?

你想让我们三个人为你做什么?

本句中，three 作 us 的同位语。

【例】Only two students, John and I, here been there before.

只有两个学生，约翰和我，以前去过那儿。

本句中，John and I 作 two students 的同位语。由于 two students 在句子中作主语，故其同位语中的人称代词要用主格形式 I 而不用宾格形式 me。

2）同位语的种类

同位语可分为限制性同位语与非限制性同位语两种。

（1）限制性同位语。

限制性同位语是指词或词组的意义受其同位语的限制。在口语中，限制性同位语的前后无须一个短暂的停顿，在书面语中也无须用符号分隔。请看下例:

【例】They each have a computer at home.

他们每人家中都有一台计算机。

【例】We Chinese are working hard to make our country rich and strong.

中国人民正努力工作使我们的国家繁荣富强。

【例】The students all went to the lecture yesterday.

昨天学生们都去听报告了。

【例】The boy looked at the two cakes and then ate them both.

那个男孩看了一下那两块蛋糕，然后把它们都吃了。

注意：all、both 及 each 作主语的同位语时，它们放在谓语动词之前，但放在第一个助动词 is、are、was、were 之后。当助动词 is、are、was、were 在句子末尾时，all、both 及 each 则要放在它们之前。请看下例:

【例】The people in this institute are all well-educated.

这个研究所的人都受过良好的教育。

【例】They have both been to America.

他们两个人都去过美国。

【例】Some of the girls are not undergraduates, but the boys all are.

有些姑娘不是大学生，但所有的小伙子都是。

（2）非限制性同位语。

非限制性同位语不限制与之同位的词或词组的意义。在口语中，非限制性同位语的前后有

一个短暂的停顿，在书面语中也用一个或一对逗号把它与其同位的词或词组分开。请看下例：
【例】His mother, a very famous English teacher, is very kind to us.
她的母亲，一位非常著名的英语教师，对我们很好。
【例】Tom was talking to Miss Wang, an undergraduate of Beijing University.
汤姆正在与王小姐谈话，王小姐是北京大学的学生。
【例】She asked for that difficult work, the one that nobody else seemed to want to do.
她要求做那个困难的工作，那个别人都不想做的工作。

若非限制性同位语前面有介绍性的词（如 that is、namely、that is to say）时，这些介绍性的词后面要用逗号，并且用括号或破折号将整个同位语与句子的其他部分隔开。若该句以同位语结尾，则用一个破折号。请看下例：
【例】He has only one thing to do——That is, to have a good sleep.
他现在只有一件事要做，那就是好好睡一觉。
【例】Two men——namely, Tom and Peter——have been employed.
两个人，即汤姆和彼德，被雇用了。

若非限制性同位语位于句子的中间，且相当长或其中有标点时，也要用一对破折号将它与句子其他部分隔开。请看下例：
【例】All kinds of printers——dot matrix printers, letter-quality printers, pin printers, bubble jet printer and laser printer——are on display.
各种各样的打印机——点阵打印机、字符打印机、针式打印机、喷墨打印机和激光打印机——都在展示。
【例】Bob and Peter——the hardest working students, according to their teacher——get high grade.
鲍博和彼德，老师认为是最用功的学生，得了高分。

2. 插入语

插入语通常是对一句话作一些附加的解释。常用来作这类附加成分的结构有：I think、I hope、I suppose、I guess、you know、don't you think、it seems、you see、it is said、it is suggested 等，它们一般放在句子末尾。请看下例：
【例】This is the best printer in this company, I suppose.
我想这是这个公司最好的打印机。
【例】You should work harder, I hope.
希望你工作再努力些。
【例】He is a promising young man, don't you think?
他是一个有前途的年轻人，你说是吧？
【例】This is the laboratory which was built three years ago, you see.
你瞧，这是三年前建造的那座大楼。

插入语也可放在句子中间，请看下例：
【例】This book, you know, is well written.
这本书，你知道的，写得很好。
【例】Her design, I think, is the best of all.

我认为她的设计是最好的。

【例】The men in that company, it is said, are all very capable.

据说那个公司的人都很能干。

在问句中有时也可以有这种插入语。请看下例:

【例】What do you think we should do now?

你认为我们现在该怎么办?

【例】When do you suppose they will finish the task?

你认为他们何时能完成这项工作?

【例】Where did she say she would buy a scanner?

她说要到哪去买扫描仪?

但若把插入语放在句首,则句子结构就会发生变化,插入语就会成为句子的主要成分,而原来句子的主要成分就会变成一个从句。请看下例:

【例】I think he must be an engineer.

我想他一定是个工程师。

本句中,I 是主语,think 是谓语,he must be an engineer 是一个宾语从句。

【例】She suggested we should complete that project as soon as possible.

她建议我们尽快完成那项工程。

【例】It is said the factory which makes CPU is over twenty years old.

据说这家制造 CPU 的工厂有二十多年的历史。

本句中,the factory which makes CPU is over twenty years old 是一主语从句。

Exercises

一、根据课文内容,判断以下叙述的正误。

(1) Wireless network refers to all kinds of computer network that uses wireless (usually, but not always) radio waves for network connections.

(2) A wireless local area network (WLAN) links only two devices over a short distance using a wireless distribution method.

(3) Mesh networks can "self heal", automatically re-routing around a node that has lost power.

(4) In a cellular network, all cells use the same set of radio frequencies from all their immediate neighbouring cells.

(5) Users can't designate a certain space which the network will be able to communicate with other devices through that network.

(6) It is costly to install physical network mediums such as TPs, coaxes, or fiber optics.

(7) For homeowners, Ethernet is less effective than wireless technology for sharing printers, scanners, and high speed internet connections.

(8) Network Elements (NEs) must be stand-alone systems.

(9) Reliable wireless service depends on the network elements at the physical layer.

(10) The more dispersive the medium is, the better total bandwidth generally.

二、根据课文内容填空。

(1) Wireless telecommunications networks are generally implemented and administered using ____ ________________.

(2) WPANs stand for ___________________________.

(3) Wireless LAN is often used in __________to connect networks in two or more buildings without ____________________.

(4) Wireless wide area networks are wireless networks that typically cover large areas, such as _________________________________or ______________________.

(5) The GSM network is divided into three major systems. They are _______________________, _______________________, and _______________________________.

(6) PCS is a ____________that can be used by ________________in North America and South Asia. __________________happened to be the first service to set up a PCS.

(7) D-AMPS stand for _______________________________.

(8) 4G technology stands for ____.

(9) Wireless networks require as few as _____________________________via __________.

(10) The maximum data rate of any single wireless link can be described by the Shannon's theorem which is related to __________________, and ______________________.

三、从供选择的词汇中选择最合适的填在文中相应数字处。

Interest in personal computer has usually focused on the individual's use of a so-called ____1____machine. But with——in a few years, this may be less important than the fact that users around the country cab be linked together over telephone lines in a network.

What's the advantage over simple making a call? Voice transmission is excellent for conveying emotional states, or limited amounts of hard ____2____, but if you have ever listened as someone described a graphic image over the phone, you quickly realized that a picture is worth much more than a thousand ____3____. And if anyone has tried to engage you in a complex discussion involving a lot of text or a great many figures, you probably found yourself postponing the conversation until you had a copy of the information in front of you. Furthermore, telephone calls must be arranged at a time suited to both parties. And you cannot just "put information out there". You must call a particular person.

Computer ____4____have none of these disadvantages. You can transmit text or graphics at your convenience; the receiver can review the information at his you can put information out onto a network "bulletin board" and whoever is interested (including people you don't know) can pick it up and use it, it communicate back to you.

Computer ____5____and specialized user groups of all sorts are springing up. Small computers may ultimately be as ubiquitous "as ubiquitous" as telephones, because they are as useful as telephones--for communicating with other people.

供选择的答案:

A. data

B. words

C. programs

D. networks

E. standard

F. stand-alone

G. integration

H. storage

I. conferencing

J. debug

四、阅读下列英文短句，选择合适的答案。

(1) When designing a system, the prime objective is to ____________________.

A. Design the cheapest system regardless of how efficient it may be

B. Design the most efficient system regardless of its cost

C. Design the most efficient system and economical system consistent with management's objectives and requirements

D. Design the system to utilize the most current technology

(2) When is queuing theory most useful?

A. The service time exhibits a large mean standard deviation

B. The service time and arrival rate can be expressed as probability distribution

C. The service facility has unlimited capacity

D. The number of service facilities exceeds the mean standard deviation

(3) A transportation problem can be solved by which of the following?

A. Non-linear programming

B. Dynamic programming

C. Simplex method

D. Zero-sum method

(4) The sample size must be increased——to double the precision of an estimate.

A. Two times

B. Four times

C. Eight times

D. Sixteen times

(5) What features should be used to evaluate a compiler?

A. Diagnostic

B. Compatibility

C. Level of implementation

D. All of the above

(6) What should be considered in evaluating computer hardware?

A. Throughput

B. Reliability

C. Back-up equipment

D. All of the above

(7) Programmers and analysts should demonstrate

A. Aptitude

B. Intelligence

C. Competence

D. All of the above

(8) Which kind of documentation is usually not kept current?

A. General information manual

B. Program documentation

C. User's manual

D. Operations manual

(9) What should be evaluated in addition to the manufacturer's hardware?

A. Training available

B. System software support

C. Ease of conversion

D. All of the above

(10) Project management involves four basic steps

A. Examine the information requirements of the managers involved

B. Select a project leader

C. Review and approval of systems design by management

D. Design the system to fulfill the information needs

The most logic sequence for executing these steps is:

① A-B-D-C

② B-A-D-C

③ B-C-A-D

④ A-C-B-D

五、听短文，在画线处填写所听到的单词或词组。

A computer virus is ____1____ that can copy itself and infect a computer without permission or knowledge of the user. The term "virus" is also commonly used, albeit erroneously, to refer to many different types of ____2____ and adware programs. The original virus may modify the copies, or the copies may ____3____ themselves, as occurs in a metamorphic virus. A virus can only spread from one computer to another when its host is taken to the uninfected computer, for instance by a user sending it over a network or the Internet, or by carrying it on a ____4____ such as a floppy disk, CD, or USB drive. Meanwhile viruses can spread to other computers by infecting files on a ____5____ or a file system that is accessed by another computer. Viruses are sometimes confused with ____6____ and ____7____ A worm can spread itself to other computers without needing to be transferred as part of a host, and a Trojan horse is a file that appears harmless. Worms and Trojans may cause

harm to either a computer system's hosted data, ____8____, or networking throughput, when executed. In general, a worm does not actually harm either the system's hardware or software, while at least in theory, a Trojan's payload may be capable of almost any type of harm if executed. Some can't be seen when the program is not running, but as soon as the ____9____ is run, the Trojan horse kicks in. That is why it is so hard for people to find viruses and other malware themselves and why they have to use ____10____ programs and registry processors.

六、计算机软件水平考试真题自测（系统分析师级）：选择填空。

WLANs are increasingly popular because they enable cost-effective connections among people, applications and data that were not possible, or not cost-effective, in the past. For example, WLAN-based applications can enable fine-grained management of supply and distribution ____A____ to improve their efficiency and reduce ____B____. WLANs can also enable entirely new business processes. To cite but one example, hospitals are using WLAN-enabled point-of-care applications to reduce errors and improve overall ____C____ care. WLAN management solutions provide a variety of other benefits that can be substantial but difficult to measure. For example, they can protect corporate data by preventing ____D____ through rogue access points. They help control salary costs, by allowing IT staffs to manage larger networks without adding staff. And they can improve overall network management by integrating with customers' existing systems, such as OpenView and UniCenter. Fortunately, it isn't necessary to measure these benefits to justify investing in WLAN management solutions, which can quickly pay for themselves simply by minimizing time-consuming ____E____ and administrative chores.

供选择的答案：

A.	（1）chores	（2）chains	（3）changes	（4）links
B.	（1）personnel	（2）expenses	（3）overhead	（4）hardware
C.	（1）finance	（2）patient	（3）affair	（4）doctor
D.	（1）intrusion	（2）aggression	（3）inbreak	（4）infall
E.	（1）exploitation	（2）connection	（3）department	（4）deployment

Skill Training

计算机英语新词的构成特点及其翻译

1. 计算机专业英语词汇的特点与构造

计算机英语的新词中，只有极少数是全新的，绝大多数都是由现有的词汇通过某种方法构造出来的。常见的构造方法有以下几种。

1）新赋意义

给公共英语中普通词汇在计算机专业语境中赋予新的、有技术特色的意义，使之成为专业术语。例如，“memory”的常用意思是“记忆”，而在计算机专业英语中给它赋予新的意

思“内存”，特指计算机的一种硬件。又如，“bus”在计算机中的意思是“总线”，成为一个技术意味极强的新词。

2）复合构造

由两个或两个以上单词构成一个新词。例如，online（在线）、offline（离线）、database（数据库）、software（软件）及 hardware（硬件）等。

3）派生法构词

在词根前面加前缀或在词根后面加后缀，从而构成一个与原单词意义相近或截然不同的新词，这个方法称为派生法。例如，可以通过给词根加前缀“un”构造出 unauthorized（*a.* 未经授权的；未经许可的）、uncheck（*vt.* 不选定，不检验，不检查）、undefined（*a.* 未下定义的，不明确的，模糊的）、undiagnosed（*adj.* 未诊断的）、undo（*vt.* 撤销）及 unlock（*v.* 解锁，开放）等；加“re”可构造出 rechargeable（*adj.* 可再充电的）、redo（*v.* 重做）、reestablish（*vt.* 重新建立）、refresh（*v.* 刷新，更新）、rekey（*v.* 再次输入）、rename（*vt.* 更名，改名，重新命名）、restart（*vt.* &*vi.* 重新启动）、restore（*vt.* 恢复，修复）及 retransmission（*n.* 转播，中继）等；以 er 结尾的词有 compiler（*n.* 编译器）、printer（*n.* 打印机）等；以 or 结尾的 calculator（*n.* 计算器）、editor（*n.* 编辑）、processor（*n.* 处理器）等。

除了以上方法外，还可以用两个或多个单词截开拼接出新词。例如，取出单词 modulate（调制）的前三个字母 mod 与单词 demodulate（解调）的前三个字母 dem，再把两个 d 字母合并，拼接为 MODEM（调制解调器）。又如，Centrino（迅驰，处理器名）的名字由来于 Center（中央）和 Neutrino（微中子）的组合。

4）专业词组

由专业单词组成固定的搭配，表达特定的专业意义。例如，variable name（变量名）、virtual machine（虚拟机）、voice recognition（语音识别）、video card（视频卡，显卡）及 Token Ring（令牌网）等。

5）大量使用缩略词

计算机英语的一个显著特点是用单词首尾字母组成一个新词，在构词法中这叫作首尾字母缩略法。缩略词可以节省大量的篇幅，但也给不知道这些缩略词的读者带来困难。例如，ISP（Internet Service Provider，因特网服务提供商）、PCI（Peripheral Control Interface，外围设备控制接口）、UML（Unified Modeling Language，统一建模语言）、CPU（Central Process Unit，中央处理器）、CSS（Cascading Style Sheets，层次式样表）、WAN（Wide Area Network，广域网）、LAN（Local Area Network，局域网等都属于缩略词。

2. 计算机专业英语词汇的翻译方法

计算机英语词汇汉译的方法多种多样，常用的方法有以下几种。

1）直译（literal translation）

所谓直译就是译出原文的字面意义。对有些英语新词语而言，汉语中有与之对应意义的词汇，或者已经约定俗成，可以采用完全直译的方法。例如，把“Microsoft”翻译为“微软”、Oracle 翻译为“甲骨文”。类似的例子还有：computer firewall（计算机防火墙）、spreadsheet（电子表格）、high-level language（高级语言）、keyboard（键盘）、client server（客户服务器）、script（脚本）、shade（阴影，底纹）等。

在计算机英语的汉译方法中，直译法是一种最常用的方法，特别是在翻译复合词时，大都采用直译法。直译法简单易行，便于理解。

2）意译（liberal translation）

当一个英语词汇没有完全对应的汉语词汇或者按照英文的字面意思直接翻译不符合汉语的表达方式时，应该意译。用汉语表达出英语词汇的含义，而不是照字面意思“硬译”。例如，mouse 原指“老鼠”，而在计算机学科中却用来指一种指点式输入设备，因此译作“鼠标”。类似的还有：tree（子目录）、laptop（笔记本电脑）、path（路径）、boot（启动）、cache（高速缓冲存储器）、click（点击，单击）、cursor（光标）、debug（调试）、run（运行）及 host（主机）等。

意译时需要翻译者具有良好的计算机专业知识，深刻地理解原文的含义，方能准确翻译。

3）音译（transliteration）

所谓音译，就是按照英语词汇的读音，翻译为相应的汉语词汇。例如，把 Twitter 翻译为“推特”。在最初阶段，这项服务只是用于向好友的手机发送文本信息，是“推送”信息的服务，此处“推特”中的“推”字尤为传神。又如，将 cracker 翻译为“骇客”。因为 cracker 就是从事恶意破解商业软件、恶意入侵别人的网站等事务，让人恐惧。如果翻译为“破解者、攻入者”，就不够形象、生动。

音译大多用于公司名称及产品名称的翻译。例如，Pentium 是 Intel 公司生产的一种计算机微处理芯片，在它刚面世时的运行速度最快，将其翻译为“奔腾”，不仅与原文发音相似，也在意义上也体现了该处理器高速快捷的性能，可谓形神俱佳。类似的翻译还有：E-mail（伊妹儿，电子邮件）、Bolg（博客）、Topology（拓扑）、Athlon（速龙，旧译阿斯龙，处理器名）、Core（酷睿，处理器名）、Google（谷歌，公司名）、Cisco（思科，公司名）及 Adobe（奥多比，公司名）、Novell（诺威，公司名）等。

4）音意兼译法（combination of literal and liberal translation）

将音译和意译两种翻译手法结合使用，兼顾语音和词义，翻译时既取其义又取其音。把 MicroBlog 翻译为“微博”，这里“微”取“Micro”之义，“博”取“Blog”之音。又如把 Internet 翻译为“因特网”，取“Inter”之音、取“net”之义。类似的还有：cyberspace（塞博空间）、Ethernet（以太网）及 softcopy（软拷贝）等。

音意兼译方法如果运用得当，译文往往新颖、易读易记。

Reading Material

IoT

1. Internet of Things

The Internet of Things (IoT)[1] is a scenario[2] in which every thing has a unique identifier and the ability to communicate over the Internet or a similar wide-area network (WAN).

---

[1] Internet of Things (IoT) 物联网

[2] scenario [siˈnɑːriəu] *n.* 情景

The technologies for the Internet of Things are already in place. Things, in this context, can be people, animals, servers, applications, shampoo bottles, cars, steering wheels, coffee machines, park benches or just about any other random item that comes to mind. Once something has a unique identifier, it can be tagged, assigned a uniform resource identifier (URI)[1] and monitored over a network. The Internet of Things is an evolutionary[2] outcome of the trend towards ubiquitous computing, a scenario in which processors are embedded in everyday objects.

Although the concept wasn't named until 1999, the Internet of Things has been in development for decades. The first Internet appliance was a Coke machine at Carnegie Melon University in the early 1980s. Programmers working several floors above the vending machine[3] wrote a server program that tracked how long it had been since a storage column in the machine had been empty. The programmers could connect to the machine over the Internet, check the status of the machine and determine whether there would be a cold drink awaiting them.

Kevin Ashton, cofounder and executive director of the Auto-ID Center at MIT, first mentioned the Internet of Things in a presentation he made to Procter & Gamble. Here's how Ashton explains the potential of the Internet of Things:

"Today computers—and therefore, the Internet—are almost wholly dependent on human beings for information. Nearly all of the roughly 50 petabytes (a petabyte is 1,024 terabytes) of data available on the Internet were first captured and created by human beings-by typing, pressing a record button, taking a digital picture or scanning a bar code[4]…The problem is, people have limited time, attention and accuracy[5]—all of which means they are not very good at capturing data about things in the real world…If we had computers that knew everything there was to know about things—using data they gathered without any help from us—we would be able to track and count everything, and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best."

IPv6[6]'s huge increase in address space[7] is another factor in the development of the Internet of Things. According to Steve Leibson, who identifies himself as "occasional docent at the Computer History Museum", the address space expansion means that we could "assign an IPV6 address to every atom on the surface of the earth, and still have enough addresses left to do another 100 + earths." In other words, we could easily assign an IP address to every thing that we wanted to monitor.

2. RFID

RFID (Radio Rrequency IDentification)[8] is a technology that incorporates the use of

[1] uniform resource identifier (URI) 统一资源标识符
[2] evolutionary [ˌiːvəˈluːʃənəri] *adj.* 进化的
[3] vending machine 自动贩卖机
[4] bar code 条形码
[5] accuracy [ˈækjurəsi] *n.* 精确性，正确度
[6] IPv6 是 Internet Protocol Version6 的缩写，译为“因特网协议第 6 版”
[7] address space 地址空间
[8] RDIF (Radio Frequency Identification) 射频识别

electromagnetic or electrostatic coupling in the radio frequency (RF)[1] portion of the electromagnetic spectrum[2] to uniquely identify an object, animal, or person. RFID is coming into increasing use in industry as an alternative to the bar code. The advantage of RFID is that it does not require direct contact or line-of-sight scanning. An RFID system consists of three components: an antenna[3] and transceiver[4] (often combined into one reader) and a transponder[5] (the tag). The antenna uses radio frequency waves to transmit a signal that activates the transponder. When activated, the tag transmits data back to the antenna. The data is used to notify a programmable logic controller that an action should occur. The action could be as simple as raising an access gate or as complicated as interfacing with a database to carry out a monetary transaction. Low-frequency RFID systems (30 kHz to 500 kHz) have short transmission ranges (generally less than six feet). High-frequency RFID systems (850 MHz to 950 MHz and 2.4 GHz to 2.5 GHz) offer longer transmission ranges (more than 90 feet). In general, the higher the frequency, the more expensive the system.

3. NFC

Near Field Communication (NFC)[6] is a short-range wireless connectivity standard (Ecma-340, ISO/IEC 18092) that uses magnetic field[7] induction[8] to enable communication between devices when they're touched together, or brought within a few centimeters of each other. Jointly developed by Philips and Sony, the standard specifies a way for the devices to establish a peer-to-peer (P2P)[9] network to exchange data. After the P2P network has been configured, another wireless communication technology, such as Bluetooth[10] or Wi-Fi, can be used for longer range communication or for transferring larger amounts of data.

Here are some examples of how NFC can be used:

- You could take pictures with a cell phone with a built-in camera, and touch an enabled computer or television set to transmit the images for display;
- You could download applications or games to a handheld device[11] by touching the computer;
- In conjunction with another wireless technology, you could transfer large files between two devices, such as a laptop and a desktop, simply by touching the two together.

4. Bluetooth

Bluetooth is a telecommunications industry specification that describes how mobile phones,

---

[1] radio frequency (RF) 无线电频率
[2] electromagnetic spectrum 电磁波频谱
[3] antenna [æn'tenə] *n.* 天线
[4] ransceiver [træn'siːvə] *n.* 收发器
[5] transponder [træn'spɔndə] *n.* 转发器，应答器，发射机
[6] Near Field Communication (NFC) 近距离无线通信技术
[7] magnetic field 磁场
[8] induction [in'dʌkʃən] *n.* 感应
[9] Peer-to-Peer (P2P) 对等网络
[10] Bluetooth ['bluːtuːθ] *n.* 蓝牙
[11] handheld device 手持式设备

computers, and personal digital assistants (PDAs) can be easily interconnected using a short-range wireless connection. Using this technology, users of cellular phones[1], pagers, and personal digital assistants can buy a three-in-one phone that can double as a portable phone[2] at home or in the office, get quickly synchronized[3] with information in a desktop or notebook computer, initiate the sending or receiving of a fax, initiate a print-out, and, in general, have all mobile and fixed computer devices totally coordinated.

Bluetooth requires that a low-cost transceiver chip be included in each device. The transceiver transmits and receives in a previously unused frequency band of 2.45 GHz that is available globally (with some variation of bandwidth in different countries). In addition to data, up to three voice channels[4] are available. Each device has a unique 48-bit address from the IEEE 802 standard. Connections can be point-to-point[5] or multipoint. The maximum range is 10 meters. Data can be exchanged at a rate of 1 megabit per second (up to 2 Mbps in the second generation of the technology). A frequency hop[6] scheme allows devices to communicate even in areas with a great deal of electromagnetic interference. Built-in encryption and verification[7] is provided.

5. Wi-Fi

Wi-Fi (short for "wireless fidelity") is a term for certain types of wireless local area network[8] (WLAN) that use specifications in the 802.11 family. The term WiFi was created by an organization called the Wi-Fi Alliance, which oversees tests that certify product interoperability. A product that passes the alliance tests is given the label "Wi-Fi certified" (a registered trademark).

Originally, Wi-Fi certification was applicable only to products using the 802.11b standard. Today, Wi-Fi can apply to products that use any 802.11 standard. The 802.11 specifications are part of an evolving set of wireless network standards known as the 802.11 family. The particular specification under which a Wi-Fi network operates is called the "flavor" of the network. Wi-Fi has gained acceptance in[9] many businesses, agencies, schools, and homes as an alternative to a wired LAN. Many airports, hotels, and fast-food facilities offer public access to Wi-Fi networks. These locations are known as hot spots[10]. Many charge a daily or hourly rate for access, but some are free. An interconnected area of hot spots and network access points is known as a hot zone[11].

Unless adequately protected, a Wi-Fi network can be susceptible to[12] access by

---

[1] cellular phone 蜂窝电话
[2] portable phone 便携式电话
[3] synchronization [ˌsiŋkrənaiˈzeiʃən] *n.* 同步
[4] voice channel 语音通道
[5] point-to-point 点对点
[6] frequency hop 跳频
[7] verification [ˌverifiˈkeiʃən] *n.* 确认，认证
[8] wireless local area network 无线局域网
[9] gain acceptance in 获得公认，得到赞同
[10] hot spot 热点
[11] hot zone 热区
[12] be susceptible to：对……敏感，可被……

unauthorized[1] users who use the access as a free Internet connection. The activity of locating and exploiting security-exposed wireless LANs is called war driving. An identifying iconography, called war chalking, has evolved. Any entity that has a wireless LAN should use security safeguards such as the Wired Equivalent Privacy (WEP)[2] encryption standard, the more recent Wi-Fi Protected Access (WPA)[3], Internet Protocol Security (IPsec), or a virtual private network (VPN)[4].

参 考 译 文

无 线 网 络

无线网络是指任何使用无线连接到网络的计算机网络（无线网络通常使用无线电波，但并不全是使用无线电波）。

家庭、通信网络和企业（商业）使用这种方法，就无须把网线引入到建筑物中或用网线连接不同位置的设备，因而节省了费用。无线通信网络使用无线电通信来运行和管理。这些运行都在 OSI 模型结构的物理层。

1. 无线网络的种类

1.1 无线 PAN

无线个人区域网（WPAN）连接一个相对较小的区域内的设备，其范围一般在一个人可及的范围之内。例如，蓝牙无线通信和不可视红外线提供 WPAN 用于连接耳机与笔记本电脑。ZigBee 也支持 WPAN 应用。Wi-Fi PAN 已经司空见惯（2010），设计师们将其整合到各种消费类电子设备中。英特尔的“我的无线网络”和 Windows 7 的“虚拟无线网络”功能，使 Wi-Fi PAN 的使用更简单，也更容易建立和配置。

1.2 无线 LAN

无线局域网（WLAN）使用无线分布方法连接两个或多个相距较近的设备，通常通过一个因特网接入点相连。使用扩频域 OFDM 技术，可以让用户在一个本地覆盖区域内移动并一直连接到该网络。

使用 IEEE802.11 无线局域网标准的产品都在 Wi-Fi 名下销售。固定无线技术实现了计算机或网络之间的点对点远程链接，通常使用专用微波或调制的激光光束沿可视路径通信。它经常用在城市中实现两个或两个以上建筑物的联网，无需安装线路连接。

1.3 无线网状网络

无线网状网络是一种无线网络，它由网状拓扑中的无线节点组成。每个节点为其他节点

[1] unauthorized [ˈʌnˈɔːθəraizd] *adj.* 未被授权的，未经认可的
[2] Wired Equivalent Privacy（WEP）有线等效保密
[3] Wi-Fi Protected Access（WPA）无线保护访问，Wi-Fi 网络安全访问
[4] virtual private network（VPN）虚拟专用网络

转发消息。网状网络可以“自愈”，当一个节点失能后，自动重新建立路由。

1.4 无线 MAN

无线城域网是一种连接多个无线局域网的无线网络。

WiMAX 是一种无线城域网，符合 IEEE 802.16 标准。

1.5 无线 WAN

无线广域网通常覆盖较大区域，如周边城镇和城市或者城市和郊区之间。这些网络可以连接公司的各个分支机构或作为一个公共因特网接入系统。无线接入点之间的连接通常在 2.4 GHz 频段，采用由抛物面天线发射的微波来实现点对点连接，而不是较小的网络所使用全向天线。一个典型的系统包含基站网关、接入点和无线桥接转发器。网络系统中的每个接入点也都可作为转发器。与再生能源系统（如光伏太阳能电池板或风力发电系统）相结合，可以成为独立系统。

1.6 蜂窝网络

蜂窝网络或移动网络是分布在称为蜂窝的地域的无线电网络，每个蜂窝至少有一个固定位置的收发器，称之为蜂窝基站或基站。在蜂窝网络中，每个蜂窝使用一组不同于相邻小区的无线电频率，以避免任何干扰。

当连接在一起时，这些蜂窝可以无线覆盖一个广大的地理区域。这使得大量的便携式收发器（例如，移动电话和寻呼机等）可以互相通信，并可与固定的收发器和网络中任何地方的电话通过基站通信，即便这些收发器传输过程中在多个基站间移动也不影响通信。

尽管最初用于手机通信，但随着智能手机的发展，移动电话网络除了电话会话外，也可以传输数据：

- 全球通（GSM）：GSM 网络分为三大系统：交换系统、基站系统以及运行和支持系统。手机连接到基站系统，然后连接到运行和支持系统，然后连接到交换系统，在交换系统把它的呼叫传送到需要去的地方。GSM 是最常用的标准并可用于大多数手机。
- 个人通信服务（PCS）：它是只能被北美和南亚手机使用的一个无线电波段。Sprint 首先建立了 PCS 服务。
- 数字高级移动电话服务（D-AMPS）：数字高级移动电话服务是 AMPS（高级移动电话服务）的升级版。由于技术的进步，AMPS 正在被逐渐淘汰。新的 GSM 网络系统正在取代旧的系统。

2. 使用

一些使用示例包括蜂窝电话作为日常无线网络的一部分，使个人通信很容易。另一个例子是洲际网络系统，用无线卫星实现全球通信。紧急服务也是一个例子，如警方利用无线网络高效通信。无论个人和企业在一个小办公楼内还是分布在世界各地，都可使用无线网络快速发送和共享数据。

3. 特性

3.1　概述

在一般意义上，无线网络为企业和家庭用户提供了各种各样的应用。

“现在，整个行业接受了一些不同的无线技术。每个无线技术依据不同的标准，这些标准，在 OSI 模型的物理层和数据链路层描述这些技术。这些标准指定的信号发送方法、地域范围和使用的频率等均不同。这种差异使某些技术更适合于家庭网络而其他更适合于大组织。”

3.2　性能

每个标准适用于不同地理范围，某种标准可能比另一标准更适用于实现无线网络。无线网络的性能可满足各种应用，如语音和视频。这种技术的使用还有扩充空间，如从 2G、3G 到最近的 4G 技术，4G 代表第四代手机移动通信标准。随着无线网络的普及，可以配置更先进的网络硬件和软件，更快地发送和接收更多的数据。

3.3　空间

空间是无线联网的另一个特点。当谈论到在难以布线的区域（如跨越街道或河流、整体运营但分隔两地的建筑物另一端的仓库）通信时，无线网络有许多优点。无线网络中允许用户指定一定的空间，在该空间中可以通过该网络与其他设备进行通信。也可以在家庭中创建空间，以消除杂乱的布线。该技术可以替代双绞线、同轴电缆或光纤等网络物理介质，这些物理介质也不便宜。

3.4　家庭

对业主来说，使用无线技术共享打印机、扫描仪和高速因特网连接比以太网更有效。无线局域网有助于节约安装电缆介质的花费，节省安装物理设备的时间，还可以建立连网设备的流动性。无线网络安装简单，只要求通过路由器直接连接到因特网的单一无线接入点。

3.5　无线网络元件

在物理层上的通信网络，也可包括多个互相连接的有线网络元件。在这些元件可以是独立的系统或产品，由同一制造商提供，也可以由服务提供商或系统集成商从几个不同的制造厂家采购组装。

可靠的无线服务依赖于物理层的网络元件来保护操作环境和应用程序。

尤其重要的是基站（BS）柜附近手机信号塔上的网络元件。附属硬件和天线位置以及相关的封闭器/电缆都必须有足够的强度、坚固性、耐腐蚀、耐雨淋并能抵抗预期的风吹日晒、风雪冰冻和其他天气条件。要求各个组成部分（如硬件、电缆、连接器和封闭器）应考虑到它们所连接的结构。

3.6 容量

3.6.1 连接

任何一个无线连接的最大数据率要符合香农定理，这涉及以赫兹（Hz）为单位的带宽和信道噪声。

3.6.2 网络

总的网络带宽取决于：介质分散的程度（通常介质分散率越高总带宽越好，因为它最小化了干扰）；使用多少有效频率，这些频率有多大噪声，是否使用定向天线，节点是否采用功率控制等。

附录 A　英语基本句型

1. 主语 + 动词 + 直接宾语

【例】They have already finished the work.
他们已经完成了那项工作。

【例】He often does that.
他经常这样做。

2. 主语 + 动词 + 动词不定式

【例】He promised to help me.
他答应帮助我。

【例】That company agreed to pay for it.
那家公司同意负担它的费用。

【例】They decided not to sell their software.
他们决定不卖掉他们的软件。

3. 主语 + 动词 + 名词或代词 + 动词不定式

【例】The professor asked them to look carefully.
教授要求他们仔细观察。

【例】I warned him not to smoke here.
我警告他不要在此吸烟。

【例】I'd like Tom to do it.
我愿意让汤姆做这件事。

4. 主语 + 动词 + 名词或代词 + （to be） + 补足语

【例】We consider him（to be）honest.
我们认为他很诚实。

【例】They believe this printer（to be）the best.
他们相信这台打印机是最好的。

【例】We proved this idea（to be）wrong.
我们证明这个观点是错误的。

5. 主语 + 动词 + 名词或代词 + 不带 to 的不定式

【例】We saw him go out.
我们看见他出去了。

【例】The guard wouldn't let him go in.
卫兵不让他进去。

【例】The father had the boy work harder.
父亲让那个男孩工作更努力些。

6. 主语+动词+名词或代词+现在分词

【例】They kept us waiting for a long time.
他们让我们等了很长时间。

【例】We found her working at her desk.
我们看见她在伏案工作。

【例】The boy saw his father coming this way.
那个男孩看见他父亲朝这边走来。

7. 主语+动词+宾语+形容词

【例】What he said made her uneasy.
他的话使她不安。

【例】When he came back, he found the room empty.
他回来时发现房间是空的。

【例】Don't get your hands dirty.
不要把手弄脏。

8. 主语+动词+宾语+名词

【例】We elected him chairman.
我们选他为班长。

【例】They all call him Fred.
他们都叫他弗雷德。

【例】They made him President of the Royal Society.
他们选他为皇家学会主席。

9. 主语+动词+宾语+过去分词

【例】Where do you often have your printer repaired?
你经常在什么地方修理你的打印机?

【例】Are you going to have your hair cut tomorrow?
你打算明天去理发吗?

【例】He couldn't make himself understood.
他不能使自己被理解。

10. 主语+动词+宾语+副词或副词短语

【例】Please put your books here.

请把你们的书放在这儿。

【例】He took his cat off.
他摘掉帽子。

【例】Later he showed me to the door.
后来他把我们送到门口。

【例】We employed him as a computer engineer.
我们雇用他为计算机工程师。

11. 主语 + 动词 + that 从句

【例】I hope everything is all right.
我希望一切都很好。

【例】He suggested that we should do it at once.
他建议我们马上做这件事。

【例】I think his design is the best.
我认为他的设计最好。

12. 主语 + 动词 + 名词或代词 + that 从句

【例】I told her that she was wrong.
我告诉她她错了。

【例】Please remind me that there is a meeting at six.
请提醒我六点开会。

【例】Mother often warns him that he mustn't smoke.
妈妈经常告诫他他不应该吸烟。

13. 主语 + 动词 + 连接词 + 不定式

【例】She was wondering which to buy.
她正踌躇不知该买哪一个。

【例】I really don't know what to do.
我真不知道该怎么办。

【例】You must learn how to look after yourself.
你必须学会如何照顾自己。

【例】Do you know when to start?
你知道什么时候开始吗?

14. 主语 + 动词 + 名词或代词 + 连接词 + 不定式

【例】Please show me how to operate this printer.
请给我示范一下如何操作这台打印机。

【例】That old engineer told her what to do next.
那位老工程师告诉她下一步该做什么。

【例】Could you advice me which to buy?
你能建议我买哪一个吗?

15. 主语+动词+连接词+从句

【例】Nobody knows who he is.
没人知道他是谁。

【例】You needn't care what he said.
你不必介意他说的话。

【例】I wonder whether he will come.
我不知道他是否会来。

16. 主语+动词+名词或代词+连接词+从句

【例】Could you tell me when he will be back?
你能告诉我他什么时候回来吗?

【例】He asked her where she bought the scanner.
他问她在哪买的这台扫描仪。

【例】Did he tell you what he was doing?
他告诉你他正在干什么吗?

17. 主语+动词+动名词

【例】He enjoys reading books.
他喜欢读书。

【例】The professor went on talking.
教授继续谈下去。

【例】Your bike needs repairing.
你的自行车需要修理。

【例】It began raining.
天开始下雨了。

18. 主语+动词+直接宾语+介词+间接宾语

【例】She gave the money to me.
她把钱给我了。

【例】His father bought a bike for him.
他父亲给他买了一辆自行车。

【例】Thank you for your help.
谢谢你的帮助。

【例】The heavy rain prevented him from coming.
那场大雨使他不能来了。

19. 主语 + 动词 + 间接宾语 + 直接宾语

【例】He showed me his design.
他给我看了他的设计。

【例】Would you do me a fervor?
你愿意给我帮个忙吗?

20. 主语 + 动词 + 第一宾语 + 第二宾语

【例】That will save us a lot of trouble.
那将省去我们许多麻烦。

【例】He struck the man a heavy blow.
他重击了那个男子一下。

21. 主语 + 动词 + 状语

【例】He waited for two hours.
他等了两个小时。

【例】They walked three miles.
他们步行了三英里。

【例】This house costs 200,000 yuan.
这座房子值二十万元。

22. 主语 + 动词

【例】The sun is shining.
阳光照耀着。

【例】Everybody breathes, eats and drinks.
人人都呼吸、饮食。

23. 主语 + 系动词 + 表语

【例】This book is very valuable.
这本书很有价值。

【例】She is in good health.
她的健康状况良好。

【例】His job is teaching.
他的工作是教书。

【例】This house is to let.
此房出租。

24. There + 动词 + 主语 + 状语

【例】There is a table in the room.

房间里有一张桌子。

【例】There remains one more experiment to be carried out.

还有一个实验要进行。

25. 主语 + 动词 + 介词 + 宾语

【例】He is listening to the radio.

他正在听收音机。

【例】He succeeded in passing the exam.

他成功地通过了考试。

【例】It depends on the weather.

此事要根据天气而定。

【例】He will deal with the matter.

他将处理这件事。

26. 主语 + 动词 + 动词不定式

【例】They should stop to have a rest.

他们应该停下来休息一下。

【例】How does he come to know that?

他怎么知道了那件事情?

【例】We woke to find the house on fire.

我们醒来发现房子着火了。

【例】Do you happen to know where he has gone?

你是否知道他去哪儿啦?

【例】We are to do it tomorrow.

我们明天做这件事。

【例】Nobody is to know.

不许任何人知道。

附录 B　英语单词速记法

学习英语的关键与难点之一就是记忆单词。机械地逐一记忆单词会花费大量的时间。而且在计算机行业中还不断涌现出一些新构造出来的词，如：unformat、undelete、resetup、uninstall 等，这些单词往往在字典中查不到。因此，必须学会科学地记忆单词和识别新单词。

其实，英语单词有其内在的结构规律，这就是构词法。掌握了构词法，则可达到举一反三、见词识义的学习效果。因此，掌握构词法是快速记忆英语单词的捷径。

常用的构词法有以下三种：合成法、转化法及派生法。

1. 合成法

由两个或两个以上的词合成一个新单词的构词方法就称为“合成法”。用合成法构成的词称为“复合词”。复合词可以有以下三种书写形式：连起来写（如：手册 handbook）；分开来写（如：汽车站 bus stop）；用连字符连在一起（如：人造的 man-made）。由三个以上单词构成一个复合词时常采用第三种方法。例如：现代的 up-to-date；一对一的 one-to-one；五岁的 five-year-old。合成词的前一个词常用来说明后一个词。例如：太阳浴 sunbathing；小车司机 car driver。

合成词的构成方法如下。

1）合成名词

（1）名词 + 名词。

work + shop ⟶ workshop 车间

wave + length ⟶ wave-length 波长

（2）名词 + 动名词。

machine + building ⟶ machine building 机器制造

book + learning ⟶ book learning 书本知识

hand + writing ⟶ handwriting 手写稿；书法

（3）动名词 + 名词。

waiting + room ⟶ waiting-room 候车室

building + material ⟶ building material 建筑材料

swimming + pool ⟶ swimming pool 游泳池

（4）形容词 + 名词。

short + hand ⟶ shorthand 速记

hard + ware ⟶ hardware 硬件

soft + ware ⟶ software 软件

black + board ⟶ blackboard 黑板

（5）动词 + 名词。

pick + pocket ⟶ pickpocket 小偷

break + water ⟶ breakwater 防水堤

（6）副词 + 动词。

in + put ⟶ input 输入

out + put ⟶ output 输出；产量

out + come ⟶ outcome 结果

（7）动词 + 副词。

feed + back ⟶ feedback 反馈

get + together ⟶ get-together 联欢会

stand + still ⟶ standstill 停顿

2）合成形容词

（1）名词 + 现在分词。

peace + loving ⟶ peace-loving 热爱和平的

epoch + making ⟶ epoch-making 划时代的

（2）名词 + 过去分词。

man + made ⟶ man-made 人造的

hand + made ⟶ hand-made 手工制作的

（3）形容词 + 现在分词。

good + looking ⟶ good-looking 好看的

fine + looking ⟶ fine-looking 美观的

deep + going ⟶ deep-going 深入的

（4）形容词 + 过去分词。

ready + made ⟶ read-made 现成的

（5）形容词 + 名词。

new + type ⟶ new-type 新型的

large + scale ⟶ large-scale 大规模的

（6）形容词 + 名词 + ed。

medium + sized ⟶ medium-sized 中型的

noble + minded ⟶ noble-minded 高尚的

（7）形容词 + 形容词。

red + hot ⟶ red-hot 炽热的

light + blue ⟶ light-blue 淡蓝的

dark + green ⟶ dark-green 深绿的

（8）数词 + 名词。

first + class ⟶ first-class 第一流的

three + way ⟶ three-way 三通的，三项的

（9）数词 + 名词 + ed。

four + cornered ⟶ four-cornered 有四个角的

one + sided ⟶ one-sided 单面的，片面的

（10）副词 + 现在分词。

hard + working ⟶ hard-working 勤劳的

ever + increasing ⟶ ever-increasing 不断增长的

（11）副词 + 过去分词。

well + known ⟶ well-known 著名的

newly + built ⟶ newly-built 新建的

above + mentioned ⟶ above-mentioned 上述的

（12）介词或副词 + 名词。

under + ground ⟶ underground 地下的

off + hand ⟶ off hand 即刻的

3）合成动词

（1） 名词 + 动词。

work + harden ⟶ to work-harden 加工硬化

heat + treat ⟶ to heat-treat 热处理

trial + produce ⟶ to trial-produce 试制

（2）形容词 + 动词。

safe + guard ⟶ safeguard 保卫

while + wash ⟶ whitewash 刷白

（3）副词或介词 + 动词。

over + heat ⟶ to overheat 过热

over + write ⟶ to overwrite 覆盖

up + set ⟶ to upset 推翻

under + line ⟶ to underline 在……下面画线

2. 转化法

在英语中，一些单词可以从一种词类转换为另一种词类，这叫作“转化”。转化后的词义往往与原来的词义有密切的联系。转化的方法主要有以下几种。

1）名词转化为动词

machine 机器 ⟶ to machine 机加工

time 时间 ⟶ to time 计时、定时

format 格式 ⟶ to format 格式化

2）动词转化为名词

to talk 交谈 ⟶ talk 谈话，讲话

to test 测验，检查 ⟶ test 测验、检验

to use 使用 ⟶ use 用途（注意发音不同）

to increase 增加 ⟶ increase 增加、增量（注意重音不同）

3）形容词转化为名词

mineral 矿物的 ⟶ mineral 矿物质

good 好的 ⟶ good 益处

final 最后的 ⟶ final 决赛

4）其他词转化为名词

twos and threes 三三两两

a must 必要条件

注意：一些以辅音结尾的单词转换词类后，词尾辅音的读音发生变化，有的拼写也发生变化。一些双音节词转换词类后，单词的重音也发生变化。名词的重读音节在第一个音节，而动词的重读音节在第二个音节。

在计算机英语中，单词的转化尤以名词与动词之间的转化居多。

3. 派生法

在一个单词的前面或后面加上一定的词缀来构成一个新词的方法称为“派生法”。词缀有前缀和后缀两种。

1）前缀

前缀由一个或几个字母组成，放在词根或单词之前，组成一个新单词。每一前缀都有一定的含义。前缀一般不超过五、六个字母。加了前缀的单词其词性一般不发生变化，只改变原来单词的意思。以下列举常用的前缀。

（1）a-，表示意义：无、非、不。

periodic 周期的 ⟶ aperiodic 非周期的

centric 中心的 ⟶ acentric 无中心的

（2）ab-，表示意义：离去、脱离。

normal 正常的 ⟶ abnormal 不正常的

（3）anti-，表示意义：反对、相反、防止、防治。

virus 病毒 ⟶ anti-virus 防病毒

magnetic 磁性的 ⟶ antimagnetic 防磁的

war 战争 ⟶ antiwar 反战的

（4）auto-，表示意义：自动、自身。

alarm 报警器 ⟶ autoalarm 自动报警器

rotation 转动、旋转 ⟶ autorotation 自动旋转，自动

（5）bi-，表示意义：两、二、重。

coloured 颜色的 ⟶ bicoloured 双色的

fold 倍 ⟶ bifold 两倍的

monthly 每月的 ⟶ bimonthly 两月一次的；双月刊

(6) by-，表示意义：边、侧、偏、副、非正式。

product 产品 ⟶ by-product 副产品

effect 作用 ⟶ by-effect 副作用

road 路 ⟶ byroad 小路

(7) centi-，表示意义：百分之一。

metre 米 ⟶ centimetre 厘米

grade 度 ⟶ centigrade 百分度

(8) co-，表示意义：共同、相互。

operation 操作 ⟶ co-operation 合作

run 管理 ⟶ co-run 共同管理

(9) counter-，表示意义：反。

action 作用 ⟶ counteraction 反作用

clockwise 顺时针方向 ⟶ counterclockwise 逆时针方向

(10) dis-，表示意义：否定、相反。

order 秩序 ⟶ disorder 混乱

charge 充电 ⟶ discharge 放电

appear 出现 ⟶ disappear 消失

(11) en-，表示意义：使……

large 大的 ⟶ enlarge 扩大

close 紧的 ⟶ enclose 封闭

danger 危险的 ⟶ endanger 使危险

(12) ex-，表示意义：出自、向外。

port 港口 ⟶ export 出口

change 变化 ⟶ exchange 交换

(13) in-，表示意义：不、非、无。

correct 正确的 ⟶ incorrect 不正确的，错误的

separable 可分开的 ⟶ inseparable 不可分的

visible 看得见的 ⟶ invisible 看不见的

注意：在 b、m、p 前加 im，在 l 前加 il。

possible 可能的 ⟶ impossible 不可能的

logical 逻辑的 ⟶ illogical 不合逻辑的

(14) inter-，表示意义：在……之间、在……之中。

national 民族的 ⟶ international 国际的

act 作用 ⟶ interact 互相作用

(15) kilo-，表示意义：千。

gram 克 ⟶ kilogram 千克

metre 米 ⟶ kilometre 千米，公里

(16) micro-，表示意义：微、小。

meter 米 ⟶ micrometer 千分尺；微米

switch 开关 ⟶ microswitch 微型开关

（17）milli-，表示意义：毫、千分之一。

volt 伏 ⟶ millivolt 毫伏

liter 升 ⟶ milliliter 毫升

gram 克 ⟶ milligram 毫克

（18）mis-，表示意义：误、错、坏。

fortune 运气 ⟶ misfortune 不幸

manage 管理 ⟶ mismanage 管理不善，处置失当

lead 引导 ⟶ mislead 误导

understand 理解 ⟶ misunderstand 误解

（19）mono-，表示意义：单、一。

plane 飞机 ⟶ monoplane 单翼飞机

rail 铁道 ⟶ monorail 单轨铁道

tone 音调 ⟶ monotone 单音，单调

（20）multi-，表示意义：多。

colour 颜色 ⟶ multicolour 多色的

program 程序 ⟶ multiprogram 多程序

form 形式 ⟶ multiform 多形的，多样的

（21）non-，表示意义：非、不、无。

stop 停 ⟶ nonstop 直达，中途不停

metal 金属 ⟶ nonmetal 非金属

conductor 导体 ⟶ nonconductor 绝缘体

（22）over-，表示意义：过分、在……上面；超过；压倒、额外。

production 生产 ⟶ overproduction 生产过剩

load 载 ⟶ overload 过载

head 头 ⟶ overhead 在头顶上的，在上头的

run 流 ⟶ overrun 溢出，超越

（23）poly-，表示意义：多、复、聚。

crystal 晶体 ⟶ polycrystal 多晶体

technical 技术的，工艺的 ⟶ polytechnical 多工艺的

atomic 原子的 ⟶ polyatomic 多原子的

（24）post-，表示意义：后。

war 战争 ⟶ postwar 战后的

liberation 解放 ⟶ post-liberation 解放后的

（25）pre-，表示意义：预先、在前。

heat 加热 ⟶ preheat 预热

condition 条件 ⟶ precondition 前提，先决条件

pay 付款 ⟶ prepay 预付，提前付

（26）re-，表示意义：再、重新。
run 运行 ⟶ rerun 重新运行
write 写 ⟶ rewrite 改写
setup 设置 ⟶ resetup 重新设置
print 打印 ⟶ reprint 重新打印
（27）semi-，表示意义：半。
conductor 导体 ⟶ semiconductor 半导体
automatic 自动的 ⟶ semiautomatic 半自动的
diameter 直径 ⟶ semidiameter 半径
（28）sub-，表示意义：在……底下；亚、次、分。
directory 目录 ⟶ subdirectory 子目录
way 路，道 ⟶ subway 地下道，地下铁路
head 标题 ⟶ subhead 副标题，小标题
area 区域 ⟶ subarea 分区
（29）super-，表示意义：超。
market 市场 ⟶ supermarket 超级市场
power 功率 ⟶ superpower 超功率
profit 利润 ⟶ superprofit 超额利润
highway 公路 ⟶ superhighway 超级公路
（30）tele-，表示意义：远、电。
vision 视力 ⟶ television 电视
graph 曲线图，图表 ⟶ telegraph 电报
meter 仪表 ⟶ telemeter 遥测计
（31）trans-，表示意义：转换；横过；超。
national 国家的 ⟶ transnational 跨国的，超越国界的
plant 种植 ⟶ transplant 移植
form 形式 ⟶ transform 转化，改变
personal 个人的 ⟶ transpersonal 非个人的，超越个人的
（32）ultra-，表示意义：超过、极端。
short 短的 ⟶ ultrashort 超短（波）的
red 红的 ⟶ ultrared 红外线的
speed 速度 ⟶ ultraspeed 超高速的
microscope 显微镜 ⟶ ultramicroscope 超显微镜
（33）un-，表示意义：反、不、非。
format 格式化 ⟶ unformat 反格式化
delete 删除 ⟶ undelete 反删除，恢复
install 安装 ⟶ uninstall 拆除，卸载
important 重要的 ⟶ unimportant 不重要的
（34）under-，表示意义：在……下；次于，低于；不足。
ground 地，地面 ⟶ underground 地下的

write 写 ⟶ underwrite 写于……之下

agent 代理人 ⟶ underagent 副代理人

size 尺寸，大小 ⟶ undersize 不够大

（35）vice-，表示意义：副的。

chairman 主席 ⟶ vice-chairman 副主席

manager 经理 ⟶ vice-manager 副经理

2）后缀

加在单词后面的词缀称为后缀。后缀一般不超过四个字母。后缀大多改变单词的词性，词的基本意义一般不变。例如：work 是动词，表示“工作”的意思；加-er 构成名词 worker，表示“工人”的意思；加-able 构成动词 workable，表示“可加工的”的意思。这一组词都与基本词义“工作”有关。

（1）常见的构成名词的后缀。

A. -age，表示：……场所；费用；行为或行为的结果；状态；情况

mileage 英里数

passage 通道

postage 邮费

breakage 破碎，破损

wastage 耗损

shortage 不足，短缺

advantage 利益

teenage 十几岁的时期

B. -ance，表示：性质，状况，行动，过程

abundance 丰富，充裕

ignorance 忽视

intelligence 智力

interference 干扰，干涉

C. -ant，-ent，表示：……者

assistant 助手

participant 参加者

agent 代理人

client 委托人，当事人

D. -er，表示：……者；……物；用于……的机械；……人

teacher 教师，导师

beginner 初学者

lawyer 律师

reader 读者

customer 顾客

barrier 栅栏，障碍物

cooker 厨具
washer 洗衣机
Londoner 伦敦人
E. -ese，表示：……的人
Chinese 中国人
Japanese 日本人
F. -ess，表示：女性
actress 女演员
hostess 女主人
waitress 女服务员
princess 公主
G. -ing，表示：动作；动作的结果；与某一动作有关者
engineering 工程，工程学
feeling 感觉
greeting 问候
fishing 钓鱼
H. -ism，表示：……主义；宗教；行为；……学；……术；……论；……法；……学派；具有某种特性；情况；状态
capitalism 资本主义
Christianism 基督教
heroism 英雄行为，英勇精神，英雄主义
magnetism 磁力学
hypnotism 催眠术
atomism 原子论
stimulism 兴奋疗法
cubism（艺术上的）立体派
characterism 特性
achromatism 无色，色盲
I. -ist，表示：某种主义者或某种信仰者；从事某种职业或研究的人
socialist 社会主义者
typist 打字员
scientist 科学家
J. -ivity，表示：性质；情况；状态
activity 活动性，活动
productivity 生产力，生产率
sensitivity 敏感性，灵敏度
passivity 被动性
K. -ment，表示：行为；状态；过程
development 发展

agreement 同意，协议
equipment 设备
investment 投资
requirement 需要
adjustment 调整
L. -ness，表示：性质；情况；状语
illness 病，疾病
firmness 结实，坚定
idleness 懒性
business 事物，商业
M. -or，表示：……的人
editor 编辑
inventor 发明者
visitor 来访者
doctor 医生
N. -ship，表示：情况；技能；身份；职位；极限
friendship 友谊，友好
membership 成员资格
professorship 教授身份
ownership 所有权，所有制
O. -th，表示：动作；过程；状态；性质
growth 成长，生长，增长
strength 力量
depth 深度
birth 起源，诞生
P. -tion，表示：行为的过程，结果，状况
junction 连接点
option 选择，选择权
function 作用，功能
addition 增加
elimination 消灭，排除
execution 完成，执行
suggestion 建议
situation 位置，处境
Q. -ty，表示：性质，状态，情况，构成抽象名词
safety 安全
beauty 美丽
cruelty 残忍
liberty 自由

R. -ure，表示：行为，行为的结果，状态，情况
culture 文化，教养
pressure 压力
failure 失败
flexure 弯曲
S. -y，表示：性质；状态；行为，构成抽象名词及学术名
difficulty 困难
discovery 发明，发现
possibility 可能性
philosophy 哲学
（2）常见的构成形容词的后缀。
A. -able，表示：能……的；可……的；易于……的；
adjustable 可调整的
suitable 适当的，合适的
favourable 受人喜欢的，赞成的
considerable 值得考虑的，相当大的，相当多的，值得重视的
B. -al，表示：属于……的；与……有关的
digital 数字的
decimal 小数的；十进位的
commercial 商业的
elemental 基本的，元素的
global 全球的，全面的
external 外部的
C. -ant，表示：……的；具有……性质的
abundant 丰富的，充分的
assistant 辅助的
vacant 空的
ascendant 上升的
D. -ar，表示：有……性质的；如……的；属于……的；有……的
familiar 熟悉的
similar 同样的
solar 太阳的
nuclear 核子的，原子核的
E. -ary，表示：与……有关的，属于……的
ordinary 普通的
necessary 必需的
customary 通常的，习惯的
elementary 初级的

F. -ed，表示：有……的；如……的
coloured 有色的，彩色的
gifted 有天才的
skilled 熟练的；有技能的
complicated 复杂的，难懂的
G. -en，表示：由……制成的；含有……质的；似……的
woolen 羊毛的
wooden 木制的
golden 金制的，似金的
leaden 铅制的
H. -ent，表示：具有……性质的；关于……的；有……行为倾向的
apparent 明显的
intelligent 聪明的，有智力的
frequent 频繁的，经常的
patent 专利的，特许的
I. -ful，表示：充满……的；具有……性质的；可……的
useful 有用的
hopeful 富有希望的
powerful 有力的
wonderful 奇妙的，惊人的
J. -ible，表示：易于……的；可……的
compatible 兼容的
visible 看得见的
flexible 易弯曲的
sensible 觉察的，感知的
K. -ic，表示：与……有关的；属于……的；有……特性的
academic 学术的
elastic 弹性的，灵活的
atomic 原子的，原子能的
periodic 周期的
L. -ive，表示：有……性质的；属于……的；与……有关的；有……倾向的
attractive 有吸引力的
active 积极的，主动的
expensive 昂贵的
productive 生产的，生产性的
M. -less，表示：没有；无；不
useless 无用的
harmless 无害的
careless 粗心的

wireless 无线的
N. -ly，表示：如……的；有……什么性质的
friendly 友好的
lovely 可爱的，令人愉快的
early 早的
timely 及时的，适时的
O. -ous，表示：多……的；有……的
dangerous 危险的
famous 著名的
poisonous 有毒的
enormous 巨大的
P. -some，表示：充满……的；易于……的；产生……的
tiresome 令人厌倦的
handsome 漂亮的，美观的
troublesome 令人烦恼的
laboursome 费力的，辛苦的
Q. -y，表示：多……的；有……的；有点……的
rainy 有雨的
ready 乐意的，准备好的
holy 神圣的
empty 空的，空虚的
funny 有趣的，滑稽可笑的
healthy 健康的
tricky 棘手的，难处理的
noisy 嘈杂的
（3）常见的构成动词的后缀。
A. -ate，表示：成为……；处理；使；作用
eliminate 排除，消除，消灭
circulate 循环，流通
terminate 终止
estimate 估计，估算
B. -en，表示：使成为，变成，引起
widen 加宽
darken 使变暗，变黑
strengthen 加强，巩固
weaken 削弱，变弱
C. -fy，表示：致成；使……化
simplify 简化
specify 指定

testify 证明，表明

beautify 美化

D. -ize，表示：变成；……化；实行

characterize 表示……的特性

emphasize 强调

realize 实现

industrialize 使工业化

recognize 认出，承认

optimize 完善

（4）常见的构成副词的后缀。

A. -ly，加在名词后，表示：每……地，每……时间一次地：加在形容词后，表示状态、程度、性质、方式

daily 每日的

yearly 每年的

greatly 大大地

quickly 迅速地

carefully 小心地，认真地

truly 真正的，确实的

B. -wards，表示：向……

outwards 向外

eastwards 向东

forwards 向前

upwards 向上

C. -wise，表示：方向；方式；状态

clockwise 顺时针方向

likewise 同样的

otherwise 否则

lengthwise 纵长

（5）常见的构成数词的后缀。

A. -th，表示：第……

fifth 第五

ninth 第九

B. -teen，表示：十……

thirteen 十三

eighteen 十八

C. -ty，表示：……十

twenty 二十

sixty 六十

附录C　参 考 答 案

Lesson 1

一、根据课文内容，判断以下叙述的正误。

（1）T　（2）F　（3）T　（4）F　（5）T

（6）T　（7）F　（8）F　（9）T　（10）T

二、根据课文内容填空。

（1）Central processing unit

（2）Basic input/output system

（3）the main circuit board that all of the other internal components connect to

（4）the basic software that allows the user to interface with the computer

（5）the most common way to connect additional components to the computer

（6）the primary device for entering information into the computer

（7）Floppy disk

（8）those in an Ethernet office network, connect each other

（9）provide an interface between each device connected to the computer, the CPU and applications.

（10）each instruction, the CPU

三、指出下列句子中的定语从句，然后把句子译成汉语。

（1）where we put our computer
我们放计算机的那个房间很大。

（2）who are requiring the full color capabilities of the color VGA monitor
那些要求彩色显示器具有全彩色性能的用户，将发现本彩色 VGA 显示器是完美的选择。

（3）why there are heat losses in a steam engine
你知道为什么蒸汽发动机中会有热量丢失吗?

（4）which allows electric current to flow easily
能让电流容易流过的材料叫导体。

（5）whose father works in AAA computer company
汤姆就是那个他父亲在 AAA 计算机公司工作的学生。

（6）which is very kind of them
他们已经邀请我们访问他们的国家，他们真是太好了。

（7）all of which involve heating it to very high temperatures
铁变成钢要经过多种工序，每道工序都要把它加热到很高的温度。

(8) where it is away from sunlight

你应该把计算机放在远离太阳的地方。

(9) in which many significant advances were made in both science and philosophy

17 世纪是一个在科学和哲学领域中取得重大发展的世纪。

(10) where there is a leaning tower about180feet high

伽利略住在比萨市，那里有一座 180 英尺的斜塔。

四、选择与以下各条叙述意义最接近的词汇。

(1) E (2) C (3) A (4) D (5) B

五、听句子，在画线处填写所听到的单词或词组。

(1) computer

(2) desktop computer

(3) World Wide Web

(4) local area network

(5) laptop

(6) tower cases

(7) keyboards

(8) designed

(9) parallel

(10) specification

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

(1) B (2) A (3) A (4) D (5) A

Lesson 2

一、根据课文内容，判断以下叙述的正误。

(1) T (2) T (3) F

(4) F (5) T (6) F

二、根据课文内容填空。

(1) TFT LCD, cathode ray tube (CRT)

(2) its dot pitch, millimeters

(3) the refresh rate, hertz (cycles per second)

(4) interlacing, even-numbered

(5) it uses very small amounts of electric power

(6) candelas per square meter

(7) response time

(8) the amount of time

(9) milliseconds (ms), faster transitions

(10) watts (W)

三、指出下列子句中的状语从句，并说明其在句子中的作用，然后把句子译成汉语。

（1） If you are new to Microsoft Windows 作条件状语；since it explains the basic concepts and skills you need to work with Windows successfully 作原因状语从句。

若你对 Microsoft Windows 不熟悉，那么本章也许是“用户指南”中对你最重要的一章，因为本章讲述了基本概念和成功地操作 Windows 所需要的技能。

（2） Whenever you see an ellipsis （…） after a menu command 作时间状语。

每当在菜单命令后看到省略号（……）时就会出现对话框。

（3） Where there's smoke 作地点状语

无风不起浪。

（4） Wherever we go 作地点状语。

无论走到哪里，我们都可以看见电在工作。

（5） because they are supposed to make it easier to compute with an operating system than without it 作原因状语。

操作系统的存在是因为有操作系统比没有操作系统时使计算更容易。

（6） while we were trying to phone him 作时间状语。

我们正准备给教授打电话时，他到了。

（7） so we are selling the car 作结果状语。

汽油已经太贵了，所以我们要卖掉这辆小汽车。

（8） Late as it was 作让步状语。

尽管天已经很晚了，他们还在继续工作。

（9） than I expected 作比较状语。

到会的人比我预料的少。

（10） so that we can hear what the others have to say 作目的状语。

我希望你不要说了以便能听见别人要说的话。

（11） as much from his mistakes as from his success 作比较状语。

一个好的工程师从他的错误中与从他的成功中学到同样多的东西。

（12） Now that you've come 作原因状语。

既然你来了，我们就讨论一下这个设计吧。

（13） as the manager told you 作方式状语。

按经理告诉你的去做吧!

（14） No matter how hard the work is 作让步状语。

无论这项工作多么难做，我们都将竭尽全力去完成它。

（15） unless you hurry up 作条件状语。

如果你不快点，就完不成这项工作了。

四、选择与以下各条叙述意义最接近的词汇。

（1） C　（2） A　（3） D　（4） F　（5） B　（6） E

五、听句子，在画线处填写所听到的单词或词组。

（1） components

（2） power supply unit

(3) storage

(4) Central Processing Unit

(5) motherboard

(6) compatible

(7) microprocessor

(8) function

(9) consists of

(10) initialize

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

(1) D (2) B (3) D (4) C (5) C

Lesson 3

一、根据课文内容，判断以下叙述的正误。

(1) F (2) T (3) F (4) F (5) T

(6) T (7) F (8) T (9) F (10) T

二、根据课文内容填空。

(1) the operating system

(2) to organize and control hardware and software so that the device it lives in behaves in a flexible but predictable way

(3) two, manage the hardware and software resources of the system, provide a consistent application interface

(4) keeping their operating systems flexible enough

(5) Real-Time Operating Systems, single-user, single task operating systems, single-user multi-task operating systems, multi-user operating systems

(6) control machinery, scientific instruments and industrial systems

(7) Single-user, single-task operating system

(8) Single-user, multi-tasking operating system

(9) many different users, simultaneously

(10) a set of instructions kept in the computer's ROM

三、指出下列子句中的动词不定式（短语），并说明其在句子中的作用，然后把它译成汉语。

(1) to develop this software 作主语。

他们花费了好几年时间来研制这个软件。

(2) how to test this software 作宾语。

我真的不知道如何测试这个软件。

(3) to learn C language 作宾语。

后来他发现学习 C 语言很重要。

(4) to do 作定语，修饰 three experiments。

他们还有三个实验要做。

（5） which computer to buy 作宾语。

你知道要买哪一台打印机吗？

（6） to be the best software in CAD 作主语补足语。

AutoCAD 被认为是 CAD 中最好的软件。

（7） talk with someone in his office 作宾语补足语。

我听见我们经理在办公室与人交谈。

（8） How to solve this problem 作主语。

如何解决这个问题还是一个问题。

（9） repair your printer 作宾语补足语。

他们让迈克修理你的打印机。

（10） In order to find out the virus in the computer 作目的状语。

为了找出计算机中的病毒，他工作了一整夜。

（11） to solve this problem 作程度状语。

这个工具不够先进，不能解决这一问题。

（12） to find some way 作宾语；to solve this problem 作定语，修饰 some way。

人们正在努力找到解决这一问题的方法。

（13） to hear that there is something wrong with your computer 作原因状语。

听说你的计算机出了故障，我们很抱歉。

（14） not to buy this printer 作宾语补足语。

他们告诉我们不要买这台打印机。

（15） how to improve the design 作表语。

我们现在所面临的问题是如何改进这个设计。

四、选择与以下各条叙述意义最接近的词汇。

（1） G　（2） D　（3） A　（4） H　（5） B

五、听句子，在画线处填写所听到的单词或词组。

（1） applications

（2） operating systems

（3） perform

（4） servers

（5） user interface

（6） services

（7） supercomputers

（8） multi-user systems

（9） originated

（10） responsible

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

（1） D　（2） A　（3） B　（4） C　（5） B

（6） A　（7） C　（8） C　（9） A　（10） C

Lesson 4

一、根据课文内容，判断以下叙述的正误。

（1） T　（2） F　（3） T　（4） F　（5） F

（6） T　（7） T　（8） F　（9） F　（10） F

二、根据课文内容填空。

（1） implementation of databases, routing tables to function

（2） arrays, records, references

（3） a group of elements, indexing

（4） fixed-sized arrays, dynamic arrays

（5） data structures, implement other data structures.

（6） it contains a pointer or link to another datum of the same type.

（7） Last In First Out (LIFO)

（8） emulates a tree structure with a set of linked nodes

（9） the child's parent node (or ancestor node, or superior), one parent, the length of the longest downward path to a leaf from that node, the height of the tree, the length of the path to its root

（10） a complete tree in itself.

三、指出下列句子中的现在分词（短语），并说明其在句子中的作用，然后把句子译成汉语。

（1） inspiring 作表语。

他刚才说的话非常鼓舞人心。

（2） talking and laughing 作状语。

他们又说又笑地走进房间。

（3） doing the experiment 作定语。

正在做实验的那些学生下周去北京。

（4） Being a student 作状语。

作为一个学生，他应该努力学习，通过考试。

（5） The discussion being over 作状语。

讨论结束了，他们都离开了会议室。

（6） There being no buses 作状语。

没有公共汽车，他们只好步行回家。

（7） repairing my bike 作宾语补足语。

我到家时，发现父亲正在修理我的自行车。

（8） surrounding a magnet 作定语。

有磁铁环绕的地方叫磁场。

（9） Not having done it right 作状语。

由于没有做对，他又试了一遍。

（10）moving 作定语；increasing its speed 作状语。

我们已经调整了机器的运动机构，这样就大大提高了其速度。

（11）moving 作定语。

一切运动的物质都有能量。

（12）walking out of the manager's office 作宾语补足语。

有人看见他从经理办公室出来。

（13）This current changing 作状语。

若电流发生变化，磁场也将发生变化。

（14）Being cooled in the air 作状语。

在空气中冷却后，这个工件就变硬了。

（15）crossing the street 作状语。

他告诉那男孩过马路要小心。

（16）leaving room for File Manager window 作状语。

把组图标放在适当位置或重新接受新程序项的组窗口的大小，使它占用桌面的一边，这样就可给文件管理器窗口留下空间。

（17）containing Excel 作定语。

要这样做，则需确保含有 Excel 的目录在 AUTOEXEC. BAT 路径中。

（18）requiring the full color capabilities of the colour VGA 作定语。

那些要求彩色显示器具有全色彩性能的用户，将发现 VGA 彩色显示器是完美的选择。

四、选择与以下各条叙述意义最接近的词汇。

（1）H　（2）D　（3）F　（4）B　（5）A

五、听短文，在画线处填写所听到的单词或词组。

（1）format

（2）data

（3）array

（4）record

（5）accessed

（6）programming

（7）algorithms

（8）elements

（9）indexing

（10）storage

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

（1）C　（2）B　（3）B　（4）A　（5）D

（6）D　（7）D　（8）B　（9）D　（10）D

Lesson 5

一、根据课文内容，判断以下叙述的正误。

（1）T　（2）T　（3）F　（4）F

（5）T　（6）F　（7）F　（8）T

二、根据课文内容填空。

（1）imperative，declarative

（2）system software，application software

（3）testing，analyzing，coordinating with other programmers

（4）imperative and declarative

（5）declarations，expressions，statements，a datatype

（6）any implementation detail

（7）functional languages and logical languages.

（8）a process.

（9）generic classes，prototypes，templates，aspects，code generators

（10）system software，application software

三、指出下列句子中的过去分词（短语），并说明其在句子中的作用，然后把句子译成汉语。

（1）built into the display 作定语。

置入显示器的若干功能提供了方便而又舒适的视觉效果。

（2）followed by some of his students 作状语。

教授来了，身后跟着他的一些学生。

（3）called a resident monitor 作定语。

为达到这一目的，建立了一个小程序，叫作驻留检测程序。

（4）moved 作表语。

他们被他所说的话深深地打动了。

（5）repaired 作宾语补足语。

约翰将要把这台打印机送去修理一下。

（6）Given more time 作状语。

若给更多的时间，他们将会做得更好。

（7）Caught in the rain 作状语。

他淋了雨，全身都湿透了。

（8）The experiment done 作状语。

做完实验后，他们全都离开了。

（9）Homework finished 作状语。

做完作业后，他跑出教室去踢足球。

（10）Made of plastics 作状语。

由于是塑料做的，所以这台机器很轻。

（11） Cooled in the air 作状语。
在空气中冷却后，这种钢变得越来越硬。
（12） filled with excitement and curiosity 作状语。
那天晚上，我们到达了他的房子，心里充满了激动和好奇。
（13） tired 作表语。
那个很长的报告令人厌倦，我们都厌烦它。
（14） his hand crossed under his head 作状语。
他正躺在床上，手交叉放在头下面。
（15） held last week 作定语。
他在上周召开的会议上，作了一场令人鼓舞的讲演。

四、从下列句子中选择 5 句正确的句子。

1. （3） 2. （5） 3. （8） 4. （9） 5. （11）

五、听短文，在画线处填写所听到的单词或词组。

（1） developer
（2） engineer
（3） specialist
（4） programming
（5） environment
（6） maintain
（7） instructions
（8） innovations
（9） elevated
（10） descriptions

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

（1） D （2） D （3） B （4） A （5） D
（6） ①D ②A ③A （7） C （8） B

Lesson 6

一、根据课文内容，判断以下叙述的正误。

（1） F （2） F （3） F （4） F （5） T
（6） T （7） T （8） T （9） F （10） F

二、根据课文内容填空。

（1） in the early1970s
（2） make a system call or call a library （CLIB） function.
（3） /bin/cc and/com/cc
（4） /＊， ＊/
（5） letters, digits, the dollar sign and the underscore
（6） a letter or an underscore

(7) 32

(8) zero

(9) auto，register，static and extern

(10) three

三、指出下列句子中的 V + ing 是动名词还是现在分词，并说明其在句子中的作用，然后把句子译成汉语。

(1) Selecting more than one item in a directory 动名词短语作主语；extending a selection 动名词短语作主语补足语。
在目录中选择多个项目被称为扩充选择。

(2) naming or renaming files and directories 动名词短语作宾语。
许多文件管理器的任务涉及命名或重新命名文件及目录。

(3) speaking to us 动名词短语作宾语。
我们不明白他为什么避免与我们谈话。

(4) leaving the door open 现在分词短语作状语。
汤姆一到家就睡着了，门还开着。

(5) Beginning some activities 动名词短语作主语；doing it 动名词短语作介词宾语。
开始一些活动就是着手做。

(6) being asked 动名词短语作介词宾语。
那个男孩不请自到。

(7) coming over 现在分词短语作宾语补足语。
当她看见他走来时，她的脸红了。

(8) telling anyone 动名词短语作介词宾语；coming back 动名词短语作介词宾语。
我不想告诉任何人他回来了。

(9) Waiting outside 现在分词短语作状语。
在外面等时，他觉得又累又担心。

(10) moving 现在分词短语作定语。
他们被那部令人激动的电影深深地感动了。

(11) hearing the news 动名词短语作介词宾语。
听到这个消息，他感到很不安。

(12) swimming 动名词作定语。
他们学校有一个游泳池。

(13) asking him for help 动名词短语作主语。
请他们帮助是没有用的。

(14) living 动名词作定语。
几年前他们的生活条件很差。

(15) finishing their work 动名词短语作介词宾语。
做完工作后，他们离开了办公室。

(16) having been sold out 现在分词短语作状语。
由于所有的票都卖完了，我们心烦意乱地往回走。

四、从供选择的词汇中选择最合适的填在文中相应数字处。

（1） D　（2） G　（3） H　（4） I　（5） E

五、听短文，在画线处填写所听到的单词或词组。

（1） cross-platform

（2） imperative

（3） developed

（4） Bell

（5） UNIX

（6） system

（7） application

（8） architectures

（9） compilers

（10） extension

六、计算机软件水平考试真题自测（程序员级）：单项选择题。

（1） B　（2） D　（3） A　（4） B　（5） A

（6） C　（7） B　（8） A　（9） A　（10） D

Lesson 7

一、根据课文内容，判断以下叙述的正误。

（1） F　（2） T　（3） T　（4） T　（5） F

（6） T　（7） T　（8） F　（9） T　（10） F

二、根据课文内容填空。

（1） both a programming language and a platform

（2） the hardware or software environment in which a program runs

（3） a software-only platform that runs on top of other hardware-based platforms

（4） The Java Virtual Machine/Java VM; the Java Application Programming Interface/Java API

（5） applets, applications

（6） API

（7） a standalone program that runs directly on the Java platform

（8） Web servers, proxy servers, mail servers, and print servers

三、把下列句子改成倒装句。

（1） Had the design been ready, we should have started working yesterday.

（2） Complicated as the problem is, a computer can work it out in a few seconds.

（3） Not until it was dark did the manager leave his office.

（4） Off went the horse.

（5） Without steel there would be no trains, no ships, nor cars or plans. No would there be electrical motors, telephones or televisions.

(6) In no way can we consider ordinary air as a good conductor.

(7) Only if you practice all the four skills of listening, speaking, reading and writing do you learn a foreign language well.

(8) Hardly had I arrived when I had a new problem to cope with.

(9) Seldom had she met with this kind of trouble.

(10) On the bed lay a beautiful young girl.

四、从供选择的词汇中选择最合适的填在文中相应数字处。

(1) D　(2) I　(3) G　(4) L　(5) B

五、听短文，在画线处填写所听到的单词或词组。

(1) Java

(2) core

(3) syntax

(4) object

(5) bytecode

(6) virtual

(7) class

(8) specifications

(9) free

(10) implementations

六、计算机软件水平考试真题自测（高级程序员级）：选择填空。

(1) C　(2) D　(3) A　(4) B　(5) A

Lesson 8

一、根据课文内容，判断以下叙述的正误。

(1) T　(2) T　(3) F　(4) T

(5) F　(6) F　(7) T　(8) F

二、根据课文内容填空。

(1) bibliographic, full-text, numeric

(2) E. F. Codd at IBM, 1970.

(3) Structured Query Language

(4) interactive queries for information, for gathering data for reports

(5) select, insert, update, a programming interface

(6) a database manager, create and access data

(7) the Structured Query Language (SQL), the object-oriented database management system (ODBMS)

(8) a distributed database

(9) a data structure for a single piece of data, records

(10) a table, a specific entity

三、用 it 改写下列句子。

（1） It is nice to hear that you have learned computer.

（2） It is really unnecessary for you to worry.

（3） It was very clear that Tom repaired the printer.

（4） It is no longer possible to buy this old kind of computer.

（5） It was not DOS but windows that he liked.

（6） It is viruses that worry most computer men.

（7） Is it a difficult job to type?

（8） It is not particularly hard to learn BASIC language or FoxBASE.

（9） She found it very tiring to find out the buy in the program.

（10） It was yesterday that this work was finished.

四、选择与以下各条叙述意义最接近的词汇。

（1） G　（2） H　（3） I　（4） C　（5） E

五、听短文，在画线处填写所听到的单词或词组。

（1） collection

（2） structure

（3） database model

（4） relational model

（5） network model

（6） organize

（7） database management system

（8） categorized

（9） query languages

（10） performance

六、计算机软件水平考试真题自测（高级程序员级）：选择填空。

（1） C　（2） D　（3） A　（4） B　（5） D

Lesson 9

一、根据课文内容，判断以下叙述的正误。

（1） T　（2） T　（3） F　（4） T　（5） F

（6） F　（7） T　（8） F　（9） F　（10） T

二、根据课文内容填空。

（1） Infrastructure-as-a-Service（IaaS）, Platform-as-a-Service（PaaS）, Software-as-a-Service（SaaS）.

（2） on demand, elastic, the provider

（3） provide easy, scalable access to computing resources and IT services

（4） Amazon Elastic Compute Cloud（EC2）, IBM's Blue Cloud, Sun Cloud, Google AppEngine and Windows Azure Services Platform.

(5) hosted services to a limited number of people behind a firewall.

(6) pays for extra compute resources when they are needed

(7) both on-premises resources, off-site (remote) server-based cloud infrastructure

(8) "X as a service," "anything as a service" or "everything as a service.", the essence of

(9) Software as a Service (SaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS).

(10) Platform as a Service (PaaS)

三、翻译下列句子。

(1) 关闭电源时，打印机保持这一中心位置。

(2) 计算机由许多被称为硬件的部件组成。

(3) 在许多系统上，软盘驱动器被称为 A 盘驱动器，硬盘驱动器被称为 C 盘驱动器。

(4) 若确保已给这个新文件指定了一个独特的名字，按回车键。

(5) 在可以使用网络工作之前，必须指定网络用户。

(6) 所有网络信息都存储在文件服务器的硬盘上。存储那些信息的系统称为“文件夹结构”。

(7) 若发现错误，程序员将暂停程序，检查内存和寄存器，直接从控制台上调试程序。

(8) 用［/q］开关可加快格式化过程。

(9) Windows 的命令列在菜单上。

(10) FOR… NEXT 循环可以嵌套；即一个 FOR…EXT 循环可以放在另一个 FOR…NEXT 循环中。

四、从供选择的词汇中选择最合适的填在文中相应数字处。

(1) B (2) C (3) E (4) A (5) E

五、听短文，在画线处填写所听到的单词或词组。

(1) machines

(2) intelligent agents

(3) engineering

(4) reasoning

(5) Communication

(6) General intelligence

(7) control theory

(8) logic

(9) data mining

(10) speech recognition

六、计算机软件水平考试真题自测（高级程序员级）：选择填空。

(1) B (2) C (3) B (4) D (5) A

Lesson 10

一、根据课文内容，判断以下叙述的正误。

(1) T (2) F (3) F (4) T (5) T

(6) T　(7) F　(8) F　(9) T　(10) T

二、根据课文内容填空。

(1) a set of related programs located at a network gateway server

(2) in a specially designated computer, no incoming request

(3) a network point that acts as an entrance to another network

(4) the computers of Internet users and the computers that serve pages to users; the computers that control traffic within your company's network or at your local Internet service provider

(5) that determines the next network point to which a packet should be forwarded toward its destination

(6) any gateway; one network meets another

(7) interfaces with an Asynchronous Transfer Mode (ATM) network

(8) a network bridge combined with a router

(9) arrives from one or more directions; is forwarded out in one or more other directions

(10) that channels incoming data; any of multiple input ports; the specific output port that will take the data toward its intended destination

三、选择与以下各条叙述意义最接近的词汇。

(1) A　(2) K　(3) G　(4) J　(5) E

四、从供选择的词汇中选择最合适的填在文中相应数字处。

(1) C　(2) B　(3) A　(4) E　(5) D

五、听短文，在画线处填写所听到的单词或词组。

(1) internetwork

(2) interconnection

(3) private

(4) technologies

(5) successor

(6) communications backbone

(7) spelled

(8) standardized

(9) administered

(10) transmission paths

六、计算机软件水平考试真题自测（系统分析师级）：选择填空。

A：(3)　B：(4)　C：(1)　D：(4)　E：(2)

Lesson 11

一、根据课文内容，判断以下叙述的正误。

(1) T　(2) F　(3) T　(4) T　(5) F

(6) F　(7) T　(8) F　(9) T　(10) F

二、根据课文内容填空。

（1） executable computer code, the supporting documents

（2） to define what a system should do, the constraints under which it must operate

（3） unit testing of software, integration testing, on the entire system, its intended users

（4） functional languages, declarative languages, imperative languages

（5） regression testing

（6） adaptation, correction, enhancement.

（7） requirements engineering

（8） actionable, measurable, testable

（9） software requirements analysis

（10） extensibility, robustness, reliability, fault-tolerance, security, maintainability, compatibility, modularity

三、用英语表示出下列数字。

（1） one thousand two hundred and thirty-four

（2） five hundred and sixth-seven thousand, three hundred and forty-five

（3） thirty-nine million, six hundred and thirty-two thousand, eight hundred and fifty-one

（4） one hundred and eighty-seven million, two hundred and thirty-eight thousand, four hundred and sixty-five

（5） fifty-eight per cent

（6） five six point o two

（7） the two hundred and eight-sixth

（8） the one thousand and first

（9） three and a half

（10） five-sixths

四、阅读下列英文短句，选择合适的答案。

（1） C （2） C （3） C （4） C （5） D

（6） A （7） B （8） B （9） A （10） A

五、听短文，在画线处填写所听到的单词或词组。

（1） introduction

（2） music lovers

（3） digital music

（4） distribute

（5） popularity

（6） compare to

（7） sound engineers

（8） have heard about

（9） downloading

（10） saving

六、计算机软件水平考试真题自测（系统分析师级）：选择填空。

A：(3)　B：(2)　C：(1)　D：(2)　E：(3)

Lesson 12

一、根据课文内容，判断以下叙述的正误。

(1) T　(2) F　(3) T　(4) F　(5) F

(6) T　(7) T　(8) F　(9) T　(10) T

二、根据课文内容填空。

(1) radio communication.

(2) Wireless personal area networks

(3) cities, installing a wired link

(4) between neighboring towns and cities, city and suburb

(5) the switching system, the base station system, the operation and support system

(6) radio band, mobile phones, Sprint

(7) Digital Advanced Mobile Phone Service

(8) fourth generation of cell phone mobile communications standards

(9) one single wireless access point connected directly to the Internet, a router

(10) the bandwidth in hertz, the noise on the channel

三、从供选择的词汇中选择最合适的填在文中相应数字处。

(1) F　(2) A　(3) B　(4) D　(5) I

四、阅读下列英文短句，选择合适的答案。

(1) C　(2) B　(3) C　(4) B　(5) D

(6) D　(7) D　(8) A　(9) D　(10) C

五、听短文，在画线处填写所听到的单词或词组。

(1) a computer program

(2) malware

(3) modify

(4) removable medium

(5) network file system

(6) computer worms

(7) Trojan horses

(8) functional performance

(9) infected code 10. spyware

六、计算机软件水平考试真题自测（系统分析师级）：选择填空。

A：(2)　B：(3)　C：(2)　D：(1)　E：(4)